REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT DATE	3. DATES COVERED *(From - To)*
18 - October - 2001	Final Technical Report	14 - Nov- 1997 to 14 - Feb - 2001; Oct 2001

4. TITLE AND SUBTITLE

Integration of Multiple Cues for Robust 3D Object Recognition: A Computational and Psychophysical Study with Applications

5a. CONTRACT NUMBER

5b. GRANT NUMBER

N00014 - 97 - 1 - 1076

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Farag, Aly A.

5d. PROJECT NUMBER

01PR02894 - 00

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Computer Vision and Image Processing Laboratory
Department of Electrical and Computer Engineering, University of Louisville
Louisville, KY 40292

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Reserach, Regional Office Chicago
536 South Clark St., RM 208
Chicago, IL 60605 -1588

10. SPONSOR/MONITOR'S ACRONYM(S)

ONR 245 (code)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution is Unlimited

20011023 012

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The project involves a comprehensive study of object description using multi-sensors. The study examines two basic scenarios for surface reconstruction. The first scenario provides a 2D - to - 3D mapping from images to surfaces and will include stereo, focus, zoom, vergence, shqpe from shading, and shape from texture. The second scenario uses active range finders to provide direct depth information about the object, i.e., provides a 3D - to - 3D mapping. The research focuses on the represenattion and fusion of information form differing image sources and the use of machine learning techniques to perform the fusion. Psychophysical studies conducted include invesigating the applicability of the recently introduced " quasi 2D coding hypothesis for 3D surface representation" in machine vision; and the behavioral evaluation of human performance with 3D fused imagery. The investigations of 3D surface reconstruction in human, computer and robot vision have an important applications in military, manufacturing and medical areas. As described in techincal report this research has been conducted in the Computer Vision and Image Processing Laboratory (CVIP Lab) at the University of Louisville. The "vision environment " created in the CVIP Lab enabled integration of multiple cues to sense, explore and reconstruct the environment layout. As a result , an active vision system called the CardEye was created.

15. SUBJECT TERMS

copmuter vision, human and machine perception, image fusion, neural networks, 3D surface reconstruction, active vision system

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	SAR	46	Aly A. Farag
					19b. TELEPONE NUMBER *(Include area code)* (502) 852 - 7510

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI-Std Z39-18

Technical Report

INTEGRATION OF MULTIPLE CUES FOR ROBUST 3D OBJECT DESCRIPTION: A COMPUTATIONAL AND PSYCHOPHYSICAL STUDY WITH APPLICATIONS

Dr. Aly A. Farag (PI)
Professor of Electrical and Computer Engineering
Director, Computer Vision and Image Processing Laboratory
CVIP Lab, Room 412 Lutz Hall
University of Louisville, Louisville, KY 40292
Phone (502) 852-7510. Fax (502) 852-1580
Email: *farag@cvip.louisville.edu*

ONR Grant Number: N00014-97-1-1076

Funding Period: 14-NOV-1997 THROUGH 14-FEB-2001

Collaborators:

Dr. E. A. Essock
Departments of Psychology and Ophthalmology and Visual Sciences
University of Louisville, Louisville, KY 40292

Dr. J. M. Zurada
Department of Electrical and Computer Engineering
University of Louisville, Louisville, KY 40292

Dr. Z. J. He
Department of Psychology
University of Louisville, Louisville, KY 40292

Abstract

In this project, we propose a comprehensive study for object description using multi-sensors. The study will examine two basic scenarios for surface reconstruction. The first scenario provides a 2D- to-3D mapping from images to surfaces, and will include stereo, focus, zoom, vergence, shape from shading, and shape from texture. The second scenario will use active range finders to provide direct depth information about the object, i.e., will provide a 3D-to-3D mapping. The research will focus on the representation and fusion of information from differing imaging sources and the use of machine learning techniques to perform the fusion. Psychophysical studies will include investigating the applicability of the recently introduced "quasi 2D coding hypothesis for 3D surface representation" in machine vision; and the behavioral evaluationof human performance with 3-D fused imagery.

As an application of the proposed research, and in order to evaluate the ideas proposed, we plan to create a "vision environment" that will allow the integration of multiple cues to sense, explore, and reconstruct the environment layout. The system will enable testing of the latest theories in human and machine perception, and will enable the integration of multisensor data for tracking, probing, and re-evaluating reconstructions, in order to provide an accurate assessment of the environment layout. The proposed system will be a great research and educational asset for studies in human, computer, and robot vision in the coming decade.

This project will enable the investigators to achieve two main goals: First, the grant will forge ties between vision researchers in several departments at the University of Louisville and at the University of Kentucky. Second, funding will support an interdisciplinary research effort investigating the representation of 3-D surface information; an important cutting-edge topic in both human and computer vision with important applications for military, manufacturing and medical areas. In addition, the grant will significantly improve the infrastructure for vision research at the University of Louisville by providing support for students, staff, and postdoctoral research fellows.

This technical report describes the research conducted at the Computer Vision and Image Processing Laboratory (CVIP Lab) of the University of Louisville during the funding period of this grant. In particular, we focus on the problem of 3D object reconstruction, and describe the CardEye active vision system that has been created at the CVIP Lab as a result of this project.

Contents

List of Figures

Chapter 1

Introduction

Vision is inherently three-dimensional. Because the majority of practical sensors provide only two-dimensional information (e.g., CCD Cameras), considerable research has been conducted to extract 3-D information from 2-D data. This is commonly known as the Shape from X problem in the computer vision literature, where X includes such paradigms as stereo, shape from shading, shape from motion, etc. Some sensors exist which can provide direct 3-D depth measurements (e.g., laser scanners) but they are usually limited to specialized environments and objects. Yet, creating a 3-D reconstructions (models) of an environment is an essential step in the applications of computer/machine vision. Based on these models, important tasks such as object recognition, tracking and navigation can be accomplished.

The 3D model builder (figure 1.1) consists of three phases: Data Aquisation, Data Preprocessing, and Surface Reconstruction. The data aquisation phase provides the computer with information about the physical object. The input to this phase can come from four different scenarios: stereo vision, shape from shading, 3D laser digitizer or Computerized Tomography (CT). The data preprocessing phase is incorporated in each technique, to facilitate the process of surface reconstruction. In a stereo vision system, features from a sequence of images are extracted and used in the surface fitting phase. Shape from shading estimates the depth of the image pixels based on the grey level of these pixels. The data obtained from the laser digitizer contains redundant information that has to be eliminated. The CT slices are segmented to mark the object that is needed to be reconstructed. The third phase in the 3D model builder is to fit a surface to the processed data. This phase is known in the computer vision field as trianglization or surface fitting. However, in the shape from shading technique, one may use multiple views for the same object, to get a complete description of the surface. In order to get a 3D model for the whole object, different views of the object are registered. This process of registration can be applied to the images or to the 3D model of each view.

In this report, we will focus on a number of stereo-based model building tcheniques. Identifying the same features among different views (the correspondence problem) is the main problem in stereo imaging. Several stereo approaches have been developed to tackle the correspondence problem, which can be categorized into two main categories: edge-based and area-based stereo.
In edge-based stereo, the correspondence problem is solved by matching the edge informa-

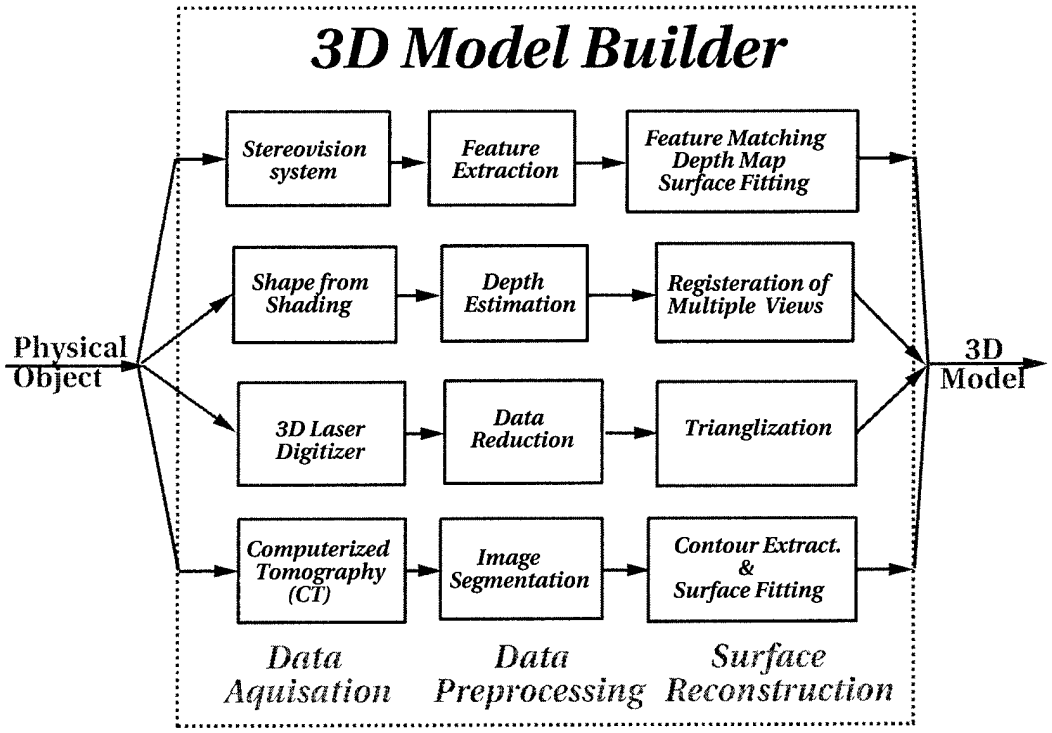


Figure 1.1: Overview of the 3D model builder.

tion in the different views. Various forms of edges have been considered such as points [16], straight line segments [4, 18], curves [48], and occluding edges [43]. This approach can generate an accurate reconstruction for the object. However, the reconstructed data is sparse and often not sufficient to build a 3-D model for the object. In area-based stereo, the matching process is applied to homogeneous grey level regions (e.g., [39]. This approach generates a dense depth map but, the reconstructed data is less accurate than the edge-based approach. A number of techniques have been developed for the integration of edge-based and region-based stereo (e.g., [15, 51, 20, 19, 17]).
As object surfaces do not often show sufficient texture (feature), stereo-based techniques are likely to fail to handle such objects. Structured light is a well-known method for introducing artificial features on surfaces. Integrating stuctured light with stereo will improve the performance of the reconstruction technique.

On the other hand, stereo unfortunately is difficult to apply to images taken from arbitrary viewpoints. Model building techniques that exploit a sequence of images taken of an object from different views have been proposed, e.g., Voxel Coloring [36], Space Carving [23] and Generalized Voxel Coloring (GVC) [9]. In this report, we also describe our apporach for model building based on the space cariving algorithm.

Two chapters of the report are dedicated to discuss stereo-based techniques and the space

carving approach. Then two applications ofthese methods are described, a trinocular active vision system used for model building and a vision system for human jaw reconstruction from a sequence of intra-oral images.

Chapter 2

Stereo-based Techniques

One way in which humans perceive depth is through a process called binocular stereopasis or stereo vision. Stereo vision uses the images viewed by each eye to recover depth information in a scene. A point in the scene is projected into different locations in each eye, where the difference between the two locations is called the disparity. Using geometric relationships between the eyes and the computed disparity value, the depth of the scene point can be calculated. Stereo vision, as used in computer systems, is similar. In stereo vision, different views of the object are acquired. By identifying the same features among these views, the depth of these features can be estimated, provided that the camera parameters are known. Identifying the same features among different views (the correspondence problem) is the main problem in stereo imaging. Several stereo approaches have been developed to tackle the correspondence problem. Figure 2.1 depicts different techniques to establish stereo corrspondences. These approaches can be categorized into two main categories: edge-based stereo and area-based stereo.

In edge-based stereo, researchers have tried to match the edges in different views. Different forms of edges have been considered such as points [16], straight line segments [4, 18], curves [48, 26], and even occluding edges [43, 35]. This approach generates an accurate construction for the object. However, the reconstructed data is sparse and it is not sufficient to build a 3D model for the object.

In area-based stereo, researchers have tried to match regions in different views. The matching process has been applied to homogeneous grey level regions (patches) assuming that the object is composed of planar patches [41, 39]. Other techniques [11] use the grey level similarity between points in the different views of the object. The grey level similarity is defined by a correlation factor that takes into consideration the grey level variations between different views of the object. This approach generates a dense depth map but, the reconstructed data is less accurate than the edge-based approach.

An integration between edge-based and area-based stereo improves the reconstruction accuracy and richness. This is what many researchers have realized and have proposed different integration approaches. Fua [14, 15] proposed to use shape from shading and stereo-based reconstruction in an iterative way to help the stereo in recovering depth information over smooth surfaces where no edges are defined. Yingen [51] starts with an edge-based stereo reconstruction and uses an area-based stereo to recover more depth information at non-edge points.

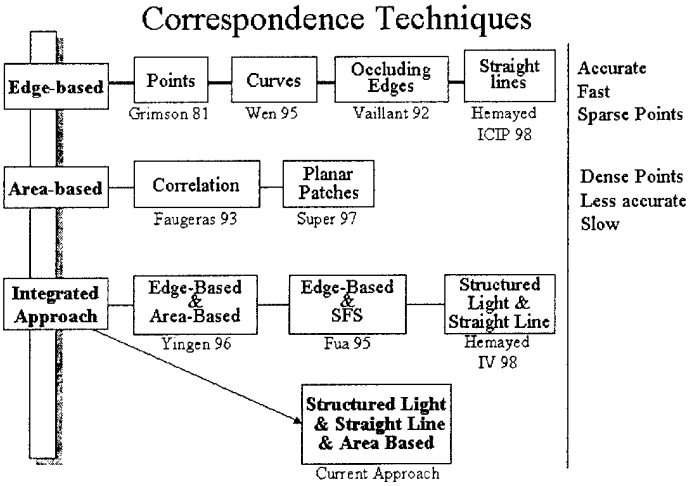


Figure 2.1: The different techniques for stereo correspondences.

Although, there are many integration approaches that utilize the merits of edge-based and area-based approaches, stereo, as a passive vision technique, is not capable of providing a robust and sufficient 3D map of the environment in all situations. The meaning of passive vision is that there is no control over the acquisition system or the lighting device. The orientation of the cameras, the zooming, or the focus cannot be changed. Controlling the acquisition system improves the image quality and as a result improves the reconstruction performance. Controlling the lighting device enable the vision system to deal with featureless objects by introducing artificial features in the scene. Controlling these parameters is what has been defined as "Active Vision" [40].

We developed a multi-stage surface reconstruction technique that is employed to handle different surface characteristics. The proposed technique integrates edge-based stereo and area-based stereo to combine the accuracy of the former and the richness of the latter, and employ the structured light to reconstruct feature-less and smooth objects that cannot be handled using edge- or area-based stereo. The integration is performed by reconstructing the actual and induced edges in the scene using the geometrical constraints of trinocular vision, followed by the application of the continuity and the epipolar constraints to grow the surface in the vicinity of the reconstructed edges. Our approach demonstrates that the integration of the three techniques: structured light, edge- and area-based stereo, enables the system to handle different surface characteristics. Fig. 2.2 shows the different modules of the multi-stage reconstruction technique.

First, three images of the scene are snapped and used as reference images $\mathbf{I_r} = \{\mathbf{I'_r}, \mathbf{I''_r}, \mathbf{I'''_r}\}$. An edge detection technique [18] is used to extract the straight line segments $\mathbf{S_r} = \{\mathbf{S'_r}, \mathbf{S''_r}, \mathbf{S'''_r}\}$ in the reference images. The pattern generator projects a laser line on the scene to create artificial features. Another three images of the scene is snapped and the introduced features are extracted using a thresholding technique. The fan out effect of the laser beam creates a blurred pattern, therefore a thinning technique is used to localize the projected pattern. The

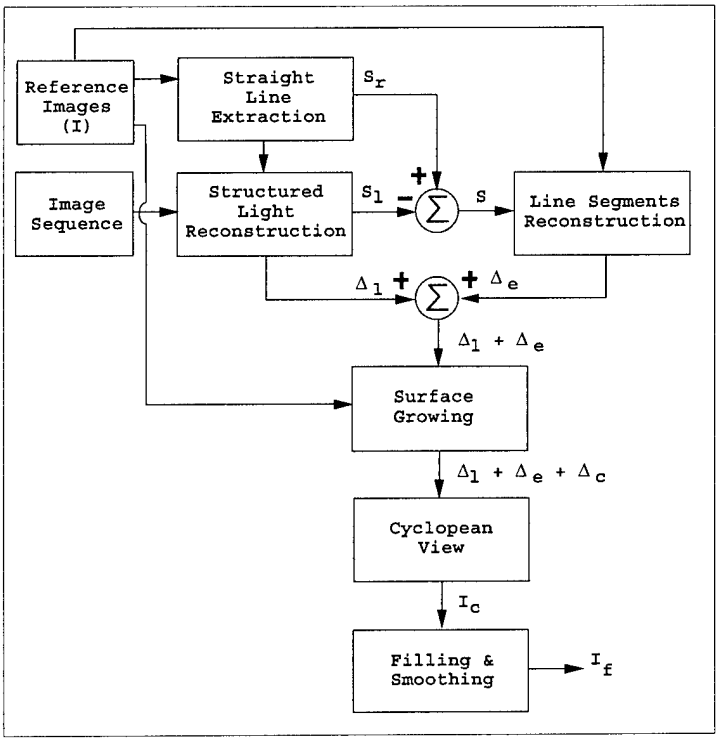


Figure 2.2: The different modules of the multi-stage surface reconstruction.

epipolar constraints are used to match the pattern from the three images and obtain a set of matched triplets. The process of projecting and matching a laser pattern is repeated for different projection planes. The orientation of the projection planes is automatically controlled and the step size is selected by the user. The sets of match obtained from each projection plan are combined in one set $\mathbf{\Delta_l}$ composed of triplets of match points $\{\mathbf{\Delta_l'}, \mathbf{\Delta_l''}, \mathbf{\Delta_l'''}\}$. A matching test is applied to the straight line segments nearby the projected patterns. The matched line segments are added to the set of match $\mathbf{\Delta_l}$. Also the matched line segments are subtracted from the extracted line segments $\mathbf{S_r}$ to obtain the residual $\mathbf{S}$ of the extracted line segments. An edge-based stereo technique is applied to the residual $\mathbf{S}$ of the extracted line segments. The technique builds another set of triplet match $\mathbf{\Delta_e}$, which is combined with the structured light set of match $\mathbf{\Delta_l}$. The combined set of match is used to guide a correlation matching technique. The correlation technique is used to match the gap points (i.e., unmatched points located between matched points along the epipolar lines) and grow a surface around the matched points. The final output is a dense set of match points $\mathbf{\Delta}$ that can be represented as a disparity map or projected onto the cyclopean camera to obtain the cyclopean view $\mathbf{I_c}$. A filling technique is used to fill the gaps that may appear in the cyclopean view.
The different modules are described in more details in the following sections.

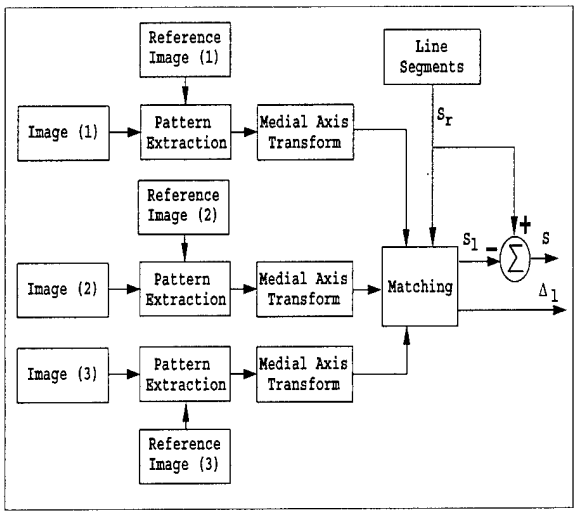


Figure 2.3: Schematic diagram of the structured light reconstruction technique

2.1 Stage I: Structured light reconstruction

As surfaces do not often show sufficient texture (feature), many reconstruction techniques rely on introducing artificial features in the scene. Structured light is a well-known method for creating such features. The basic idea of structure light method is to project a pattern of light (grids, stripes, elliptical pattern, etc.) onto an object. These patterns are distorted by the object surface. The distorted pattern is used to deduce the surface shape [44, 10, 27]. In our approach, we used a straight line laser pattern. Generally, the pattern is projected as thick lines. For proper localization of the pattern, preprocessing of the pattern is needed before reconstruction. An overview of the surface reconstruction using structured light is shown in Fig. 2.3. First, the pattern is extracted from the image using simple subtraction and thresholding techniques. The medial axis of the pattern is extracted using an automatic medial axis pruning technique [30]. The epipolar constraints are used to match the medial axis of the extracted pattern and obtain a set $\mathbf{\Delta_I}$ of match triplets $\{\mathbf{\Delta_I'}, \mathbf{\Delta_I''}, \mathbf{\Delta_I'''}\}$. In the following sections, we discuss the medial axis transform and the matching technique.

2.1.1 Medial Axis Transform (MAT)

The MAT of a shape is the locus of the centers of all maximal discs contained in the shape. A maximal disc contained in the shape is any circle with its interior that is contained in the shape (i.e., has empty intersection with the exterior of the shape) such that the circle touches the boundary of the shape at two or more points. Equivalently, for each interior point of the shape, consider the set of boundary points closest to it. Then the MAT of the shape is the set of all those interior points of the shape that have at least two closest boundary points. The medial axis extraction has three processes. The first process is extracting the object boundaries from a binary image and encoding them as a polygonal chain whose vertices are the endpoints of 'raster cracks,' the elementary straight line segments separating black and

white pixels. The second process is computing the Voronoi tessellation (Euclidean metric) of the polygonal chain. Finally, the medial axis is extracted from the Voronoi diagram [30].

2.1.2 Matching Technique

The matching technique employs the well known stereo geometric constraint, the epipolar constraint. The epipolar constraint between two points $\mathbf{m}' \in \mathbf{I}'$ and $\mathbf{m}'' \in \mathbf{I}''$ is represented in the following form: for $\mathbf{m}'$ to be a match for $\mathbf{m}''$, then the following equation should be satisfied:

$$\tilde{\mathbf{m}''}^{\mathbf{T}} \ \mathbf{F} \ \tilde{\mathbf{m}}' = \mathbf{0} \tag{2.1}$$

where $\mathbf{F}$ is the fundamental matrix that relates image $\mathbf{I}'$ to $\mathbf{I}''$. However, due to inaccuracy of the camera calibration, the previous equation does not have exactly zero on its right-hand side. The epipolar equation is modified to be

$$|\tilde{\mathbf{m}''}^{\mathbf{T}} \ \mathbf{F} \ \tilde{\mathbf{m}}'| \leq \epsilon \tag{2.2}$$

where ϵ depends on the accuracy of the measurements and camera calibration and $\tilde{\mathbf{m}}$ is the augmented vector of $\mathbf{m}$. The previous equation does not measure a physical quantity so we modify it to the following form:

$$d(m', m'') = |\frac{\tilde{\mathbf{m}''}^{\mathbf{T}} \ \mathbf{F} \ \tilde{\mathbf{m}}'}{\sqrt{l_1'^2 + l_2'^2}}| \leq \epsilon \tag{2.3}$$

where $\mathbf{l}_1', \mathbf{l}_2'$ are components of the epipolar line of $\mathbf{m}''$ which is given by $\mathbf{l}' = \mathbf{F}\mathbf{m}'' = [\mathbf{l}_1', \mathbf{l}_2', \mathbf{l}_3']$. The modified form of the epipolar constraint measures the physical distance between the point $\mathbf{m}'$ and the epipolar line $\mathbf{l}'$ of $\mathbf{m}''$. Similar formulas are constructed for the pair $(\mathbf{m}' \in \mathbf{I}', \mathbf{m}''' \in \mathbf{I}''')$ and the pair $(\mathbf{m}'' \in \mathbf{I}'', \mathbf{m}''' \in \mathbf{I}''')$. In the case of three cameras, a triplet of match $(\mathbf{m}', \mathbf{m}'', \mathbf{m}''')$ is accepted only if the epipolar constraints is fulfilled in the three pairs.

Since we are using a vertical laser line, the matching can be performed between two images only. However, we are using the third camera to improve the accuracy of the reconstruction process. We employ the match obtained from artificial features in matching actual edges. If there is an intersection between the set of match $\mathbf{\Delta_l}$ and the extracted line segments $\mathbf{S_r}$, then the line segments $\mathbf{S_l}$ pass through this intersection is in match if they fulfill the matching criteria defined in the next section. Then the matched line segment $\mathbf{S_l}$ is added to the set of match $\mathbf{\Delta_l}$ and subtracted from the extracted line segments $\mathbf{S_r}$. The matching algorithm is outlined in Algorithm 1.

Algorithm 1 An outline of the structured light matching algorithm

Input :
three sets of 2D pattern points $\mathbf{m}', \mathbf{m}'', \mathbf{m}'''$
three sets of extracted line segments $\mathbf{S}'_\mathbf{r}, \mathbf{S}''_\mathbf{r}, \mathbf{S}''_\mathbf{r}$
Output:
A set of triplet match $\mathbf{\Delta_l}$
Algorithm:
for all $\mathbf{m} = (\mathbf{m}'_\mathbf{i}, \mathbf{m}''_\mathbf{j}, \mathbf{m}'''_\mathbf{k}) \in (\mathbf{m}', \mathbf{m}'', \mathbf{m}''')$, respectively. **do**
  if $Epipolar(\mathbf{m}'_\mathbf{i}, \mathbf{m}''_\mathbf{i}, \mathbf{m}'''_\mathbf{i})$ **then**
    if $Intersect((\mathbf{m}'_\mathbf{i}, \mathbf{m}''_\mathbf{i}, \mathbf{m}'''_\mathbf{i}), (\mathbf{S}'_\mathbf{r}, \mathbf{S}''_\mathbf{r}, \mathbf{S}'''_\mathbf{r}))$ **then**
      Let $\mathbf{l}' \in \mathbf{S}'_\mathbf{r}$ be the line that passes by $\mathbf{m}'_\mathbf{i}$. Similarly $\mathbf{l}''$ and $\mathbf{l}'''$,
      if $Match(\mathbf{l}', \mathbf{l}'', \mathbf{l}''')$ **then**
        Let $\mathbf{S_l} = (\mathbf{s}'_\mathbf{l}, \mathbf{s}''_\mathbf{l}, \mathbf{s}'''_\mathbf{l}) =$ **CommonSegment**$(\mathbf{l}', \mathbf{l}'', \mathbf{l}''')$, $\mathbf{\Delta_l} = \mathbf{\Delta_l} \bigcup \mathbf{S_l} \bigcup \mathbf{m}$
        $\mathbf{S_r} = \mathbf{S_r} - \mathbf{S_l}$
      end if
    else
      $\mathbf{\Delta_l} = \mathbf{\Delta_l} \bigcup \mathbf{m}$
    end if
  end if
end for

In Algorithm 1, $Epipolar(\mathbf{m}'_\mathbf{i}, \mathbf{m}''_\mathbf{i}, \mathbf{m}'''_\mathbf{i})$ is true only if the epipolar constraints is fulfilled in the three pairs $(\mathbf{m}', \mathbf{m}'')$, $(\mathbf{m}', \mathbf{m}''')$ and $(\mathbf{m}'', \mathbf{m}''')$, $Intersect((\mathbf{m}'_\mathbf{i}, \mathbf{m}''_\mathbf{i}, \mathbf{m}'''_\mathbf{i}), (\mathbf{S}'_\mathbf{r}, \mathbf{S}''_\mathbf{r}, \mathbf{S}'''_\mathbf{r}))$ is true only if $Distance(\mathbf{m}, \mathbf{S}) \leq \epsilon$ in the three images. The distance is computed as follows:

$$Distance(\mathbf{m}, \mathbf{S}) = |\frac{\mathbf{mS}}{\sqrt{\mathbf{S}_1^2 + \mathbf{S}_2^2}}| \tag{2.4}$$

where $(\mathbf{S_1}, \mathbf{S_2}, \mathbf{S_3})$ is the normalized components of the line $\mathbf{S}$. In the $Epipolar()$ and the $Intersection()$ functions, we consider the threshold to be 0.5 pixel. The $Match()$ and $CommonSegment()$ are discussed in the next section.

2.2 Stage II: Edge-based reconstruction

Many stereo vision algorithms have been developed to estimate surfaces from stereo images of a scene acquired using a fixed, known camera configuration. The paradigm used in most of these algorithms consists of three main phases: feature detection, feature matching and depth estimation. In our case, the used features are straight line segments. The line segments, as high level features, speed up the matching process by compressing the search space. In the same time, they enable us to use the geometric features attached to them such as spatial orientation. The accuracy and the reliability of the line segment matching counterbalances the disadvantage of reconstructing few points [12].

The theme of our work is similar to Ayache's work [4]. Both fall under the category of prediction and verification techniques, as classified by Faugeras [12]. However, our approach

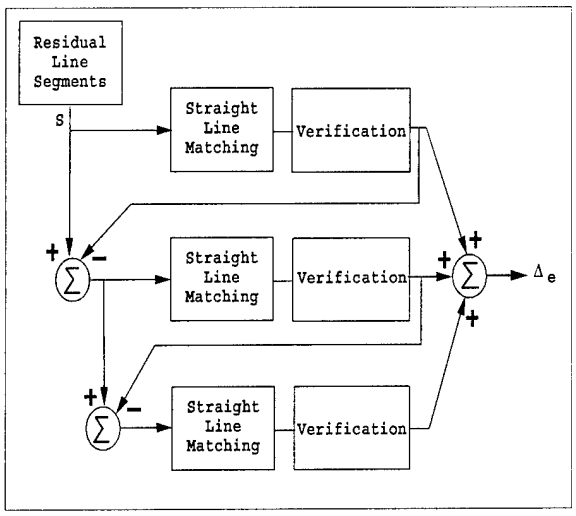


Figure 2.4: Schematic diagram of the line segments reconstruction technique

differs from Ayache's approach in three points. First, we work directly on the obtained, unrectified images to avoid the distortion results from rectifying the images. Second, we do not consider the relative length of the line segments as a matching criterion. Thus, our approach can handle the foreshortening problem. Third, we eliminate the dependency on predefined thresholds by verifying the matching using a global optimization process.

Fig. 2.4 shows an overview of our edge-based reconstruction technique. The matching process runs iteratively on the residual of the extracted line segments. The first iteration tries to match all the line segments while the other iterations consider only those segments that failed to be matched in previous iterations. The iterative process continues until no more matches are found between the segments. In our experiments, we found that three iterations were often enough for the matching. The last process of the system is to lump all the line segments together before reconstructing their end points then their corresponding 3-D line segments.

In this paper, we present a brief discussion of the edge-based stereo. More details can be found in [18]. The discussion is organized into three subsections the geometrical constraints of the trinocular vision system, the matching algorithm and the validation process.

2.2.1 The Geometrical Constrains

The trinocular vision system has rich geometrical constraints that can be used to faithfully reconstruct the 3-D environment. However, the accuracy of detected features and the calibration process weaken the importance of these constraints. In order to efficiently utilize the geometrical constraints, the uncertainties of image measurements should be taken into consideration.

In this section, we present the essential geometrical constraints that are related to straight line segments. For each constraint, we discuss the effect of the uncertainty.

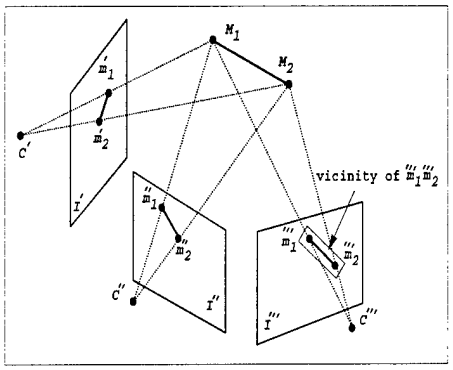


Figure 2.5: The uniqueness constraint. If $\tilde{\mathbf{m}}'_1\tilde{\mathbf{m}}'_2$ matches $\tilde{\mathbf{m}}''_1\tilde{\mathbf{m}}''_2$ then they match also $\tilde{\mathbf{m}}'''_1\tilde{\mathbf{m}}'''_2$

2.2.1.1 Uniqueness

If two 2-D lines from two images form a match, then there is at most one line in the third image that matches these two lines. The uniqueness constraint is known as the trifocal constraint [50]. As shown in Fig. 2.5, if $\tilde{\mathbf{m}}_1\tilde{\mathbf{m}}_2 \in \mathbf{I}$ matches $\tilde{\mathbf{m}}'_1\tilde{\mathbf{m}}'_2 \in \mathbf{I}'$, then $\tilde{\mathbf{m}}''_1\tilde{\mathbf{m}}''_2 \in \mathbf{I}''$ is the projection of $\tilde{\mathbf{M}}_1\tilde{\mathbf{M}}_2$ in image $\mathbf{I}''$. Due to uncertainty, the projection of $\tilde{\mathbf{M}}_1\tilde{\mathbf{M}}_2$ lies just in the vicinity of $\tilde{\mathbf{m}}''_1\tilde{\mathbf{m}}''_2$ and not at its exact location.

2.2.1.2 Projection

Every 3-D straight line segment is projected as a straight line in the image. As shown in Fig. 2.6 (left), $\tilde{\mathbf{m}}'_1\tilde{\mathbf{m}}'_2$ is the projection of $\tilde{\mathbf{M}}_1\tilde{\mathbf{M}}_2$ in image $\mathbf{I}'$. However, due to the uncertainty of feature detection, $\tilde{\mathbf{M}}_1\tilde{\mathbf{M}}_2$ may have a projection as two lines, $\tilde{\mathbf{m}}''_1\tilde{\mathbf{m}}''_3$ and $\tilde{\mathbf{m}}''_3\tilde{\mathbf{m}}''_2$, or more.

2.2.1.3 Back Projection

A 2-D line segment can be a projection of more than one 3-D line segment. As shown in Fig. 2.6 (right), $\tilde{\mathbf{m}}'_1\tilde{\mathbf{m}}'_4$ is the projection of two lines $\tilde{\mathbf{M}}_1\tilde{\mathbf{M}}_2$, $\tilde{\mathbf{M}}_3\tilde{\mathbf{M}}_4$. However, this issue is resolved by projecting the 3-D line using different views.

2.2.1.4 Epipolar Constraint

If two 2-D line segments from two images form a match, then their end points must obey the epipolar constraint. The epipolar constraint is represented in its modified form, Eq. 2.3.

2.2.2 The Matching Algorithm

The matching algorithm employs the geometrical constraints discussed above in building a set of triplets of matched line segments $\mathbf{\Delta_e} = (\mathbf{S}', \mathbf{S}'', \mathbf{S}''')$ of the residual line segments $\mathbf{S} = \{(\mathbf{S}'), (\mathbf{S}''), (\mathbf{S}'')\}$. The strategy of the algorithm is to match all the combinations of line

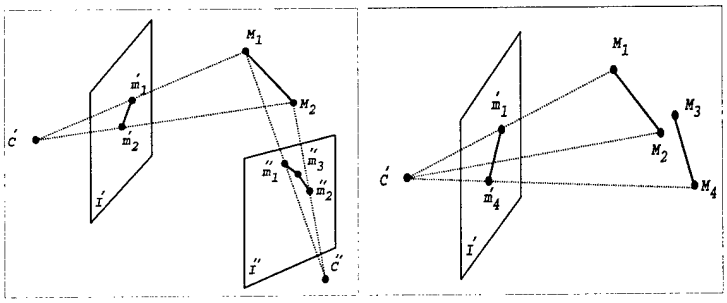


Figure 2.6: (left) The projection constraint. A 3-D line segment may be projected as one or two line segments. (right) The back projection constraint. Two 3-D line segments may be projected as one segment

segments $(\mathbf{S}', \mathbf{S}'', \mathbf{S}''')$ from the three images. A triplet of match is accepted only if it obeys the geometrical constraints. Algorithm 2 outlines the line segments matching algorithm.

Algorithm 2 An outline of the line segments matching algorithm

$Z = \phi$,
for all $\mathbf{s}' \in \mathbf{S}'$ **do**
  $\pi' = VisualPlane(\mathbf{s}', \mathbf{C}')$,
  for all $\mathbf{s}'' \in \mathbf{S}''$ **do**
    $\pi'' = VisualPlane(\mathbf{s}'', \mathbf{C}'')$,
    $\mathbf{S_3} = InterSectTwoPlanes(\pi', \pi'')$,
    $\mathbf{s_3} = Project(\mathbf{S_3}, \mathbf{C}''')$,
    for all $\mathbf{s}''' \in \mathbf{S}'''$ **do**
      if $TestMatch(\mathbf{s_3}, \mathbf{s}''')$ **then**
        $\mathbf{Z} = \mathbf{Z} \bigcup CommonSegment(\mathbf{s}', \mathbf{s}'', \mathbf{s}''')$
      end if
    end for
  end for
end for

In Algorithm 2, the functions $VisualPlane()$ and $IntersectTwoPlanes()$ are described in the Appendix. The function $Project()$ projects the end points of the line segments. The function $TestMatch(\mathbf{s_3}, \mathbf{s}''')$ performs the following test:

1. $|Angle(\mathbf{s_3}, \mathbf{s}''')| < \epsilon_1$,
2. $|Distance(\mathbf{s_3}, \mathbf{s}''')| < \epsilon_2$,
3. $CommonSegment(\mathbf{s}', \mathbf{s}'', \mathbf{s}''') \neq \phi$.

Thus, for a triplet $(\mathbf{s}', \mathbf{s}'', \mathbf{s}''')$ to be a candidate match, the corresponding line segment of $\mathbf{s}', \mathbf{s}''$ in I''', which is $\mathbf{s_3}$, should have similar orientation as $\mathbf{s}'''$ and lie in the vicinity of $\mathbf{s}'''$ and there should be a common segment between $(\mathbf{s}', \mathbf{s}'', \mathbf{s}''')$. The first two conditions results

from the uniqueness constraint in the presence of noise due to the calibration process. The common segment condition (computed in Algorithm 3) results from the epipolar constraint.

Algorithm 3 An outline of the *CommonSegment*() function

Represent the line segments by their end points, then, $\mathbf{s}' = (\mathbf{m}'_1, \mathbf{m}'_2)$, $\mathbf{s}'' = (\mathbf{m}''_1, \mathbf{m}''_2)$ and $\mathbf{s}''' = (\mathbf{m}'''_1, \mathbf{m}'''_2)$,
Let the reference image $\mathbf{I} = \mathbf{I}'$,
Let $\mathbf{p}'_1 = \mathbf{m}'_1$,
Let $\mathbf{l}''_1$ be the epipolar line of point $\mathbf{p}'_1$ in $\mathbf{I}''$,
Let $\mathbf{l}'''_1$ be the epipolar line of point $\mathbf{p}'_1$ in $\mathbf{I}'''$,
if $Intersect(\mathbf{l}''_1, \mathbf{s}'') = \mathbf{p}''_2 \neq \phi$ AND $Intersect(\mathbf{l}'''_1, \mathbf{s}''') = \mathbf{p}'''_2 \neq \phi$ **then**
  $(\mathbf{p}'_1, \mathbf{p}''_1, \mathbf{p}'''_1)$ are end points of the common segment,
else
  Let the reference image $\mathbf{I} = \mathbf{I}''$ and $\mathbf{p}''_1 = \mathbf{m}''_1$ OR
  Let the reference image $\mathbf{I} = \mathbf{I}'''$ and $\mathbf{p}'''_1 = \mathbf{m}'''_1$
  Repeat the above steps to get $(\mathbf{p}'_1, \mathbf{p}''_1, \mathbf{p}'''_1)$
end if
Repeat for $(\mathbf{p}'_2, \mathbf{p}''_2, \mathbf{p}'''_2)$

The matching algorithm starts by assuming that two lines, $\mathbf{s}' \in \mathbf{S}'$ and $\mathbf{s}'' \in \mathbf{S}''$, are corresponding to each other, which is not true. Hence, the output of this process is a set of triplets of line segments that have false and also true matches. The disambiguation of these matches is handled using the following validation process.

2.2.3 The Validation Process

The concept behind our validation process is the fact that if the end points on the line segments are matched, then their line segments are also matched. Based on this observation, the validation process is reduced to a matching task between the end points of the line segments.

The line segments are matched based on a similarity measure between their end points. The similarity measure is a correlation score computed in the neighborhood of each pair of points. For the triplet $(\mathbf{s}', \mathbf{s}'', \mathbf{s}''')$ where their end points are denoted by $(\mathbf{m}'_1\mathbf{m}'_2, \mathbf{m}''_1\mathbf{m}''_2, \mathbf{m}'''_1\mathbf{m}'''_2)$. The matching score is computed as follows:

$$Score(\mathbf{s}', \mathbf{s}'', \mathbf{s}''') = \frac{1}{2} Similarity(\mathbf{m}'_1, \mathbf{m}''_1, \mathbf{m}'''_1) + \frac{1}{2} Similarity(\mathbf{m}'_2, \mathbf{m}''_2, \mathbf{m}'''_2) \tag{2.5}$$

where

$$Similarity(\mathbf{m}', \mathbf{m}'', \mathbf{m}''') = \frac{1}{3} Cor(\mathbf{m}', \mathbf{m}'') + \frac{1}{3} Cor(\mathbf{m}', \mathbf{m}''') + \frac{1}{3} Cor(\mathbf{m}'', \mathbf{m}''') \tag{2.6}$$

$$Cor(\mathbf{m}', \mathbf{m}'') = \frac{1}{(2N+1)(2M+1)\sqrt{\sigma^2(\mathbf{I}') \times \sigma^2(\mathbf{I}'')}} \times \sum_{i=-N}^{N} \sum_{j=-M}^{M} \left[\mathbf{I}'(u'+i, v'+j) - \overline{\mathbf{I}'(u', v')}\right] \times \left[\mathbf{I}''(u''+i, v''+j) - \overline{\mathbf{I}''(u'', v'')}\right] \quad (2.7)$$

where N, M are the half length and width of correlation window, respectively, $\overline{\mathbf{I}'(u', v')}$ (Similarly $\overline{\mathbf{I}''(u'', v'')}$) is the average at point (u', v') of $\mathbf{I}'$

and is given by:

$$\overline{\mathbf{I}'(u', v')} = \frac{\sum_{i=-N}^{N} \sum_{j=-M}^{M} \mathbf{I}'(u'+i, v'+j)}{(2N+1)(2M+1)} \quad (2.8)$$

and $\sigma(\mathbf{I}')$ (Similarly $\sigma(\mathbf{I}'')$) is the standard deviation of the image $\mathbf{I}'$ in the neighborhood $(2N+1) \times (2M+1)$ of (u', v'), which is given by

$$\sigma(\mathbf{I}') = \sqrt{\frac{\sum_{i=-N}^{N} \sum_{j=-M}^{M} \mathbf{I}'^2(u'+i, v'+j)}{(2N+1)(2M+1)} - \overline{\mathbf{I}'^2(u', v')}} \quad (2.9)$$

The matching score is in the range [-1, 1], where -1 denotes a bad match and 1 denotes a very strong match.

Matching the end points is performed in two steps. In the first steps, we compute the matching score (Eq. 2.5) for each triplet of match. In the second step, the matching score is optimized subject to the uniqueness constraint. Applying the uniqueness constraint in the optimization process means forcing the matching triplets to have a unique representation of line segments. Fortunately, the search space for our task is small, and it can be an exhaustive search could be applied to guarantee optimum solution.

Based on the back projection constraint, the matching and validation process builds a set of triplets of matched line segments where each line entry can be a partial segment of the original line segment. As a consequence, we build another set of line segments that is the difference between the matched line set and the extracted line sets. The matching and validation process is applied iteratively to the new line segments set. The iteration process is finished when it fails to match new line segments.

At this stage, we have a set of triplets of matched segments $(\mathbf{S_1}, \mathbf{S_2}, \mathbf{S_3})$. Using the camera parameters, we reconstruct the 3-D correspondence of the matched line segments. The reconstruction process is simplified by constructing only the end points of the line segments. This is based on the projection constraint and the fact that straight lines are well defined by their end points.

2.3 Stage III: Surface Growing

The strategy of the surface growing process is based on the continuity constraint [12]. The basic idea of this constraint is that the world is mostly made up of objects with smooth

surfaces. This means that the reconstruction function, which assigns to a triplet of matched points a 3D point $\mathbf{M}$, is smooth almost everywhere. Thus if $\mathbf{M_1}$ and $\mathbf{M_2}$ are two 3D points with projections $(\mathbf{m}'_1, \mathbf{m}''_1, \mathbf{m}'''_1)$ and $(\mathbf{m}'_2, \mathbf{m}''_2, \mathbf{m}'''_2)$ respectively. Then if $\|\mathbf{M_1} - \mathbf{M_2}\| < \epsilon$, then $(\mathbf{m}'_2, \mathbf{m}''_2, \mathbf{m}'''_2)$ lie in the vicinity of $(\mathbf{m}'_1, \mathbf{m}''_1, \mathbf{m}'''_1)$.

We employ the continuity constraint in growing a surface around the known matched points. The algorithm is outlined in Algorithm 4. The idea of the algorithm is search for new matches nearby the known ones. Search windows are constructed around the known triplet of match. A similarity measure (Eq. 2.6) and epipolar constraint (Eq. 2.3) have been used to match new points inside the search window. The size of the search window is selected to be half the distance between successive patterns projected in the scene. This choice of window size ensures minimum overlapping scan of the entire scene. Localizing the search in small windows speeds up the search process and eliminates false matches arise from repetitive texture in the scene.

At this stage, we have a dense set of triplets of match points ($\mathbf{\Delta} = \mathbf{\Delta_l} \bigcup \mathbf{\Delta_e} \bigcup \mathbf{\Delta_c}$). Using the camera parameters, we reconstruct the 3D correspondence of the matched points [5]. The obtained 3D data could be represented in several formats depending on the user's needs. We can fit a surface to the data and then generate different views to that surface. Range images could be generated by projecting the 3D data into any of the three cameras and decoding the depth of each point into grey level. In our case, we are interesting in generating the cyclopean view, defined in the introduction section. Therefore, we project the data into the cyclopean camera and encode the depth of each point[1] as grey level. The cyclopean image is denoted by $\mathbf{I_c}$, which is a 2D array of predefined dimensions. The entries of the array are normalized between [0-255]. However, we assign the value ϕ for points that do not have 3D correspondence in $\mathbf{\Delta}$. These empty points are filled using a filling algorithm described in the following section.

2.4 Stage IV: Filling and Smoothing

The filling algorithm is another implementation of the continuity constraint. However, it is much faster than the surface growing implementation. The filling algorithm is applied to 2D data and it does not perform intensive computation functions. Algorithm 5 outlines the filling algorithm. The filling technique assigns to each empty entry of $\mathbf{I_c}$ the mean value of its non empty neighbors. The filling process is followed by a low pass filtering operation to smooth out the final cyclopean view $\mathbf{I_f}$.

2.5 Experimental Results

The previous approach has been used to reconstruct several different realistic scenes. Varieties of real object with different surface characteristics and sizes have been placed at different distances from the system. Some results are shown in Fig. 2.7. Fig. 2.7 illustrates the different stages of the reconstruction procedure as follows:

- Part (a) shows the original image of the scene,

[1] It is the z-coordinate of our system coordinate system.

Algorithm 4 An outline of the surface growing algorithm

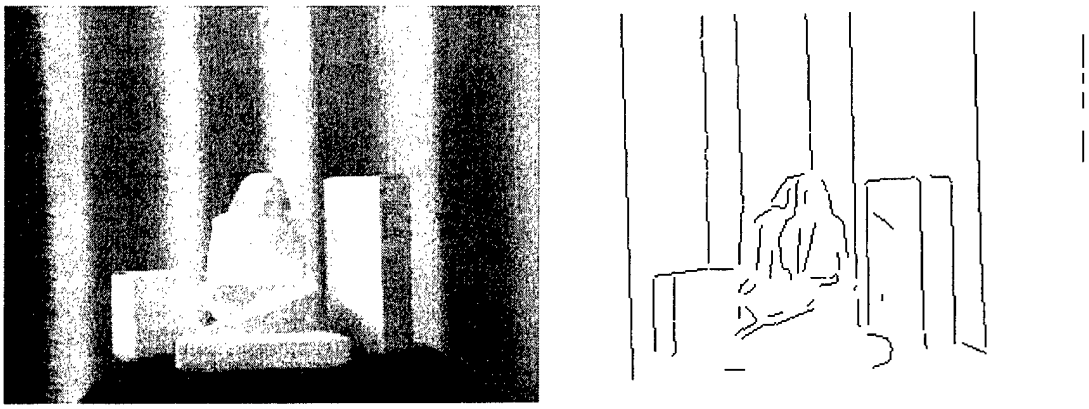
$\mathbf{\Delta_c} = \mathbf{\Phi}$
for all $\mathbf{m} = (\mathbf{m}', \mathbf{m}'', \mathbf{m}''') \in (\mathbf{\Delta_l} \bigcup \mathbf{\Delta_e})$ **do**
  Let $\mathbf{R}', \mathbf{R}'', \mathbf{R}'''$ be $N \times N$ search windows centered around $\mathbf{m}', \mathbf{m}'', \mathbf{m}'''$,
  for all $\mathbf{p}' \in \mathbf{R}'$ **do**
    $\mathbf{Z} = \mathbf{\Phi}$,
    for all $\mathbf{p}'', \mathbf{p}''' \in \mathbf{R}'', \mathbf{R}'''$ **do**
      Let $\mathbf{p} = (\mathbf{p}', \mathbf{p}'', \mathbf{p}''')$,
      if $Epipolar(\mathbf{p})$ and $Similarity(\mathbf{p})$ **then**
        $\mathbf{Z} = \mathbf{Z} \bigcup \mathbf{p}$,
      end if
    end for
    Let $N = Cardinality(\mathbf{Z})$,
    if $N == 1$ **then**
      $\mathbf{\Delta_c} = \mathbf{\Delta_c} \bigcup \mathbf{Z}$,
    else if $N > 1$ **then**
      Determine $\mathbf{z} \in \mathbf{Z}$ that maximize the similarity score,
      $\mathbf{\Delta_c} = \mathbf{\Delta_c} \bigcup \mathbf{z}$,
    end if
  end for
end for

Algorithm 5 An outline of the filling algorithm

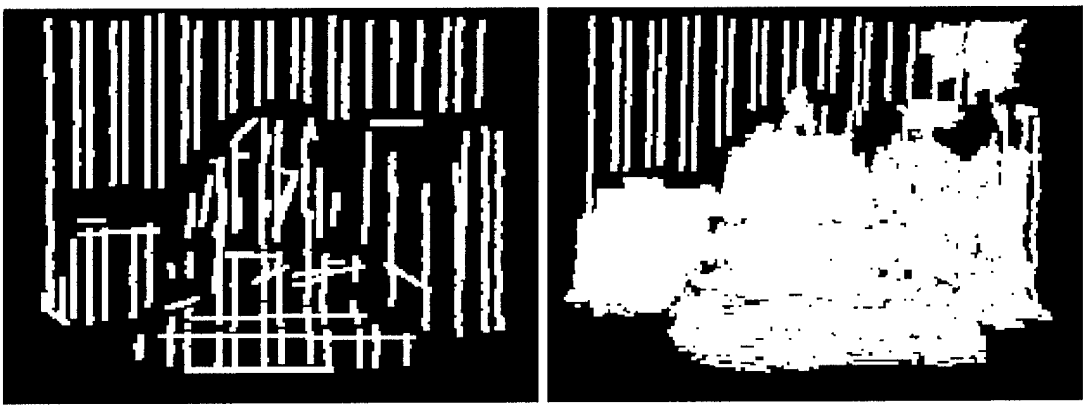
for all $(r, c) \in \mathbf{I_c}$ **do**
  if $\mathbf{I_c}(r, c) = \phi$ **then**
    Let $\mathbf{R}$ be 7×7 window centered around (r, c),
    $N = 0$, $Mean = 0$,
    for all $(i, j) \in R$ **do**
      if $\mathbf{I_c}(i, j) \neq \phi$ **then**
        $N = N + 1$,
        $Mean = Mean + \mathbf{I_c}(i, j)$,
      end if
    end for
    $\mathbf{I_c}(r, c) = \frac{Mean}{N}$,
  end if
end for

- Part (b) shows the extracted line segments in that scene,
- Part (c) shows the matched points of the structured light and the edge-based stereo, ,
- Part (d) shows the total matched points obtained after the surface growing,
- Part (e) shows the final reconstruction a range image,
- Part (f) shows the final reconstruction as a mesh.

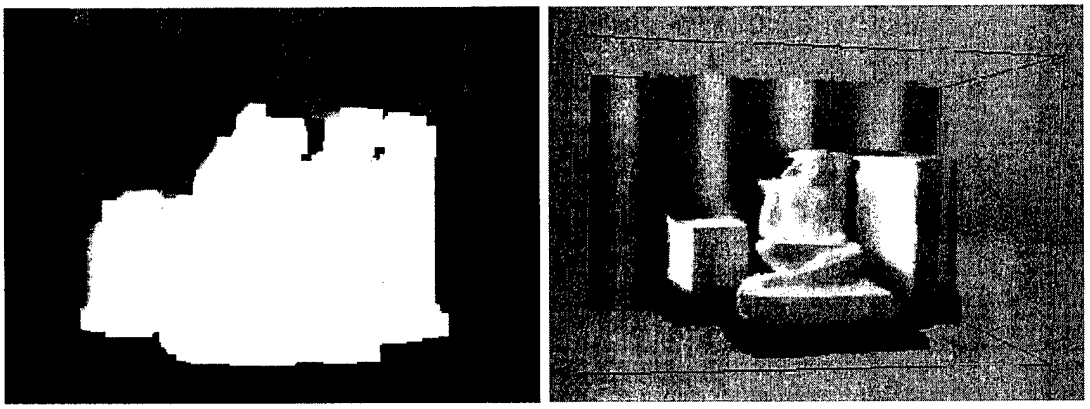
The scene in Fig. 2.7 composed of a statue, two boxes and a curtain. The statue has a smooth surface without texture and with very few edges. This type of object cannot be reconstructed with an edge-based, area-based stereo or even an integration between them. The results show how the system overcomes this problem by using the structured light. The system reconstructs the artificial features introduced by the pattern generator and use them to guide an area-based stereo technique. The edge-based stereo is used to reconstruct the actual edges and thus determines the discontinuity of the object's surface. The two boxes have long edges and an edge-based stereo works accurately in this case. However, to reconstruct the inner surface of this object, an area-based stereo is needed. As the object does not show enough texture to distinguish between its points, reconstructing more points inside the inner surface is needed to avoid the failure of area-based stereo technique. Thus, the structured light is used to support the area-based technique.



(a) (b)



(c) (d)



(e) (f)

Figure 2.7: The reconstruction results at different stages (a) The original image $\mathbf{I}$. (b) The extracted line segments S_r, (c) The matched edges before surface growing $\mathbf{\Delta_l} \bigcup \mathbf{\Delta_e}$, (d) The matched points after surface growing $\mathbf{\Delta}$, (e) The cyclopean view as a range image, (f) The cyclopean view as a mesh $\mathbf{I_f}$

Chapter 3

Reconstruction From A sequence of Images

One of the most researched topics in computer vision involves reconstructing the shape of a 3D scene from one to several images. Several different approaches have been formulated to solve this problem. The stereo approach, which has been modeled after the human vision system, has been by far the most widely used. Stereo techniques [12] find points in two or more input images that correspond to the same point in the scene (the correspondence problem). Then the depth of the scene point is determined using knowledge of the camera locations and triangulation (the depth estimation problem). Unfortunately, stereo is difficult to apply to images taken from arbitrary viewpoints. This is a two-sided problem. If the input viewpoints are far apart, then corresponding image points are hard to find automatically. On the other hand, if the viewpoints are close together, then small measurement errors result in large errors in the calculated depths. Furthermore, stereo produces a 2D depth map and integrating many such maps into a true 3D model is a challenging problem [29]. As a result, stereo is subject to several limitations. In an attempt to depart from the problems and limitations of stereo, different or variant approaches have been formulated.

Seitz and Dyer [36] have presented Voxel Coloring, a method that represents volume as a discrete collection of small cubes called voxels. Voxel Coloring uses the assumption of Lambertian surfaces and produces a color-consistent set of voxels to represent 3D objects. However in order to treat occlusion, several restrictions are imposed on the locations of the cameras.

A variation of stereo that resembles Voxel Coloring has been developed by Roy and Cox [33]. By projecting discrete 3D grid points into an arbitrary number of images, they collect color variance statistics. They impose a smoothness constraint both along and across epipolar lines. Their algorithm produces better reconstruction than conventional stereo. However, a major shortcoming is that the algorithm does not model occlusion.

Faugeras and Keriven [13] have used a variational principle that must be satisfied by the surfaces of the objects in the scene to deduce a set of partial differential equations. A level set formulation of these PDE's is used to deform an initial set of surfaces to move towards the objects to be detected. Their method can both handle an arbitrary number of images and also deal with occlusion. However, it is unclear whether or not their method will converge for every condition. Although they have produced some impressive reconstructions, they did

not provide runtime and memory statistics. Therefore we do not know if their method is very practical for reconstruction in terms of processing time and memory usage.

Like Voxel Coloring [36] of Seitz and Dyer, space carving [23] uses the idea of volumetric representation of shape and Lambertian assumption for surfaces. The main advantage of space carving over Voxel Coloring is that Voxel Coloring imposes constraints on camera locations while space carving does not.

We have developed a mechanism for object reconstruction from a sequence of images based on Space Carving [36]. The idea is to create a system that is capable of getting images of the object from different viewpoints. The system receives as input from the user the number of images it needs to acquire and the rotation angle it has to perform between the views. At the same time, the system should keep track of the camera parameters at each position, see Fig. 3.1. Typically, space carving is a computationally intensive algorithm. Reconstruction using this method has been shown to take a considerable amount of time (20 minutes or more). In order to reduce the reconstruction time, we develop a parallel version of space carving. The output of space carving is an unformated cloud of 3D points. In order to display the created 3D model on an existing 3D graphics software, we convert the output of space carving to an OOGL (Object Oriented Graphics Library) file.

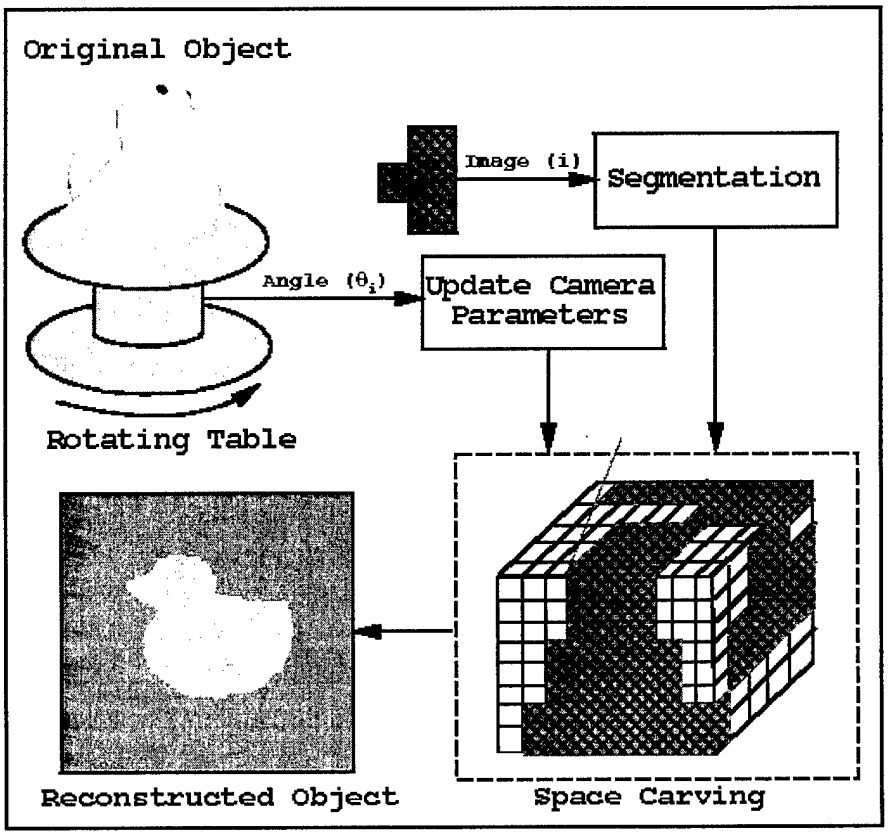


Figure 3.1: Overview of the reconstruction technique.

In the follwoing, we describe the theory and the implementation details of the approach.

3.1 Theory of Space Carving

Image formation is typically achieved by projecting a 3D shape onto a two-dimensional plane. In 3D object reconstruction, we attempt to achieve the reverse process of image formation by regenerating a 3D shape from various 2D projections. However, there is not a one-to-one relation between 3D shapes and their projections. Several 3D shapes can have the same 2D projections. As an illustration, a sphere and a cylinder can all have the same 2D projections from certain viewpoints. This ambiguity can be resolved by using more 2D projections to estimate a 3D shape. Space carving [23] attempts to produce the maximal 3D shape that is consistent with all the images.

Space carving starts with an initial volume, V, that includes the scene to be reconstructed. This 3D space is then discretized into a set of voxels. The idea is to successively carve (remove) some voxels until the final 3D shape, V^*, agrees with all the input images.

Each voxel in the initial volume is projected back to the different images using their respective projection matrices. To decide whether a voxel should be carved or not, the idea of color-consistency is used. We assume a Lambertian model for the surface of the object. Under this model, light reflected from a single point on the surface of the object has the same intensity in all directions. Therefore, for a voxel to belong to the surface of the object, it must have the same color intensity for all its projections to the different images provided. Voxels that are inconsistent with a single color, are viewed as free space in which different light rays intersect. By removing all color-inconsistent voxels, we are able to approximate a maximal photo-consistent shape that is defined by all the input images. The basic idea of space carving is illustrated in Figure 3.2. Three input images are used to generate the 3D model of the shape shown in the images. Voxels that project on the input images to pixels of similar color are kept and assigned that color. Voxels that project on the input images to pixels of different colors are removed.

3.2 Algorithm

Although the general idea in space carving is straightforward, modeling an algorithm to provide the desired results is not an easy task as the problem of occlusion must be treated. Given N input images and their respective projection matrices, the algorithm must be able to guarantee convergence to the maximal photo-consistent shape.

Let us start by defining an arbitrary volume V that contains the object. V must be discretized into a finite collection of voxels $v_1, v_2, ..., v_n$. Next, we need to determine the voxels on the surface of V. These voxels are the ones that can be visible to the different images at this point. This step is called visibility computation and consists on finding the voxels that can be visible to the images and not occluded by other voxels. Once we have computed the set of visible voxels, $Vis(V)$, each voxel, $v_i \in Vis(V)$, is projected back to the input images to see if its color is consistent in all the different images where v_i is seen. If the color is consistent, then the voxel is kept. Otherwise, it is carved. After all voxels on $Vis(V)$ have been

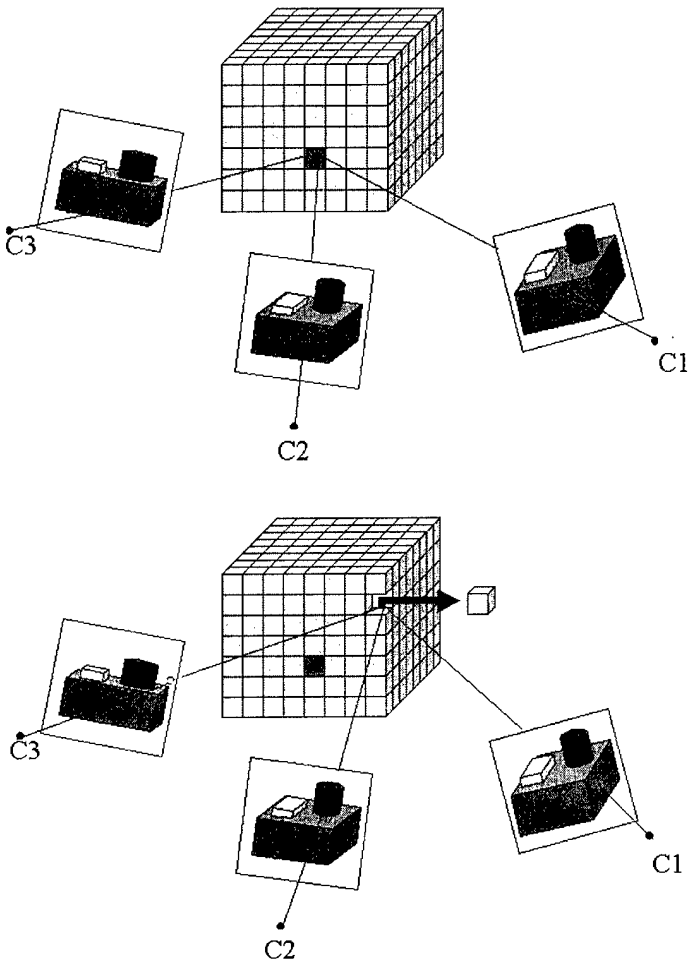


Figure 3.2: *Basic idea of space carving. Voxels are projected to the input images using their respective projection matrices. C1, C2 and C3 represent the optical centers of the three cameras. (a) Consistent voxels are assigned the color of their projections. (b) Inconsistent voxels are removed from the volume.*

tested, we update the volume V, which still contains the maximal photo-consistent shape. Once again, we compute the new set of voxels on the surface of the updated volume V and test all these visible voxels. This process is repeated until no more voxels are carved away and volume V remains unchanged. The final volume V has the maximal photo-consistent shape. The different steps of the algorithm just described are outlined below.

Space carving Algorithm.
Step 1: Initialize V to be a superset of the scene and discretize V.
Step 2:

- Determine the set of voxels $Vis(V)$ on the surface of V.

- Project each voxel v on *Vis(V)* to the different images where v is visible.
- Determine the photo-consistency of each voxel v on *Vis(V)*.

Step 3: If no non-photoconsistent voxel is found, set $V^* = V$ and terminate. Otherwise, set $V = V$ - {non-photoconsistent v's} and return to Step 2.

The most challenging task in this algorithm is Step 2. In this step, we must be able to compute the set of voxels that can be visible to the cameras each time we update the volume V by removing inconsistent voxels. Typically, visible voxels are on the surface of V and occlude other voxels. When a visible voxel is removed from the volume V, other voxels that it occluded now become visible. It is essential to keep track of voxels' visibility in a way that can be updated efficiently. This is achieved by implementing the space carving algorithm in a multi-sweep fashion.

3.2.1 Multi-Sweep Implementation

The multi-sweep implementation of space carving [23] consists of sweeping a plane through the scene volume, testing the visibility of the voxels on that plane and then determining the photo-consistency of the visible voxels. As we move the plane along a sweep direction, only images from cameras that are directed toward the sweep direction and that overlook the current plane, are used for photo-consistency check of the voxels on the plane. The plane-sweep technique can be summarized into two rules: (1) we only consider voxels on a similar plane as we move this plane along a specific direction, (2) we only use images from cameras that are located on the "front" side (i.e. side where cameras are directed towards the scene volume) of the plane to test the voxels for photo-consistency. These rules will guarantee that voxels are always tested from closer to further viewpoints of the cameras along a sweep direction. All occlusion relations are therefore captured. Typically, if a voxel $\boldsymbol{p}$ occludes a voxel $\boldsymbol{q}$ from a camera $\boldsymbol{C}$, then the sweep guarantees that $\boldsymbol{p}$ will be always visited before $\boldsymbol{q}$.

In the multi-sweep implementation, space carving is arranged to perform several passes until no photo-inconsistent voxels are found. Each pass consists of sweeping a plane through the scene volume in six directions while testing the voxels on the plane. Sweeping is performed in increasing x-coordinate, increasing y-coordinate, increasing z-coordinate, decreasing x-coordinate, decreasing y-coordinate and decreasing z-coordinate directions. Multiple sweeps rather than a single sweep are necessary in each pass of the algorithm in order to guarantee that: (i) every image is used for photo-consistency check (if a camera, directed toward the object, is located past the last plane through the scene volume along a specific sweep direction, then its image will never be used for photo-consistency check in that sweep); (ii) occlusion relations are further treated for voxels that lie on the same plane along a particular sweep direction by changing the sweep direction.

To test visibility of voxels on the plane during a sweep, we compute the equation of the optical ray passing through each voxel for each camera under consideration. If an uncarved voxel intersects the optical ray of a camera to a particular voxel being checked for visibility, then the latter voxel is declared not visible (i.e. occluded) to the camera. Otherwise, the voxel is declared visible and is checked for photo-consistency on the entire set of images on which it is visible. The multi-sweep algorithm allows us to visit voxels in an order that

makes visual information updates more efficient. To decide whether a voxel should be carved or not, we base our decision on photo-consistency. Each voxel is projected to the different images on which it is visible. The standard deviation of the intensities of the pixels the voxel projects to, is used to determine whether the voxel should be carved or not. If the standard deviation is above a certain threshold, the voxel is declared photo-inconsistent and it is carved. Otherwise, the voxel is declared photo-consistent and it is kept.

3.3 Image Pre-Processing

Space carving [23] reconstructs 3-D shape by removing photo-inconsistent voxels from a chosen initial volume that includes the object of interest. Captured images of the object include both the object and the background. In general, space carving keeps all the photo-consistent voxels to form the maximal photo-consistent shape. Some voxels that project to the background may be consistent with a single color, and therefore the reconstructed shape will include both object voxels and background voxels. If we are only interested in recovering the shape of a particular object, background voxels that are included in the final volume should be removed as well.

In order to ensure that these background voxels are never deemed consistent or left uncarved, several methods can be used. One method would be physically to alter the background of the object (for example, by placing cardboards of different color behind the object) during capture of the images. Another method would be to use a background with a homogeneous color easily differentiable from the object being reconstructed. The images can then be segmented to remove the background by applying some image processing algorithms. We choose the latter approach. The object to reconstruct is typically placed behind a black background. The background is then removed by thresholding the images of the object. Figure 3.3 shows a few captured images of a reindeer piggy bank and the resulting images after thresholding was applied to remove the background.

3.4 Parallel Implementation

To implement the space carving algorithm, we use the C language. Each voxel in the initial volume is represented by its center (x, y, z coordinates), its color and a flag that shows whether the voxel is existent or has been carved away. The space carving program ends when no photo-inconsistent voxel is found (i.e. no voxel is carved). The final output file includes only the center and color of the voxels that have not been carved or removed. The computationally intense program of space carving typically requires a lot of processing time as we have to perform multiple sweeps, in which every voxel on the plane has to be tested for visibility and then for photo-consistency on all the images on which the voxel is visible. In order to speed program execution, we develop a parallel version of space carving to run on the 24-processor Onyx R10000 supercomputer. We take advantage of the fact that no occlusion relation is assumed for the voxels on the same plane during a sweep. Therefore, visibility information can be computed independently for all the voxels on the same plane. The task of processing (computing visibility, and then testing for photo-consistency) every

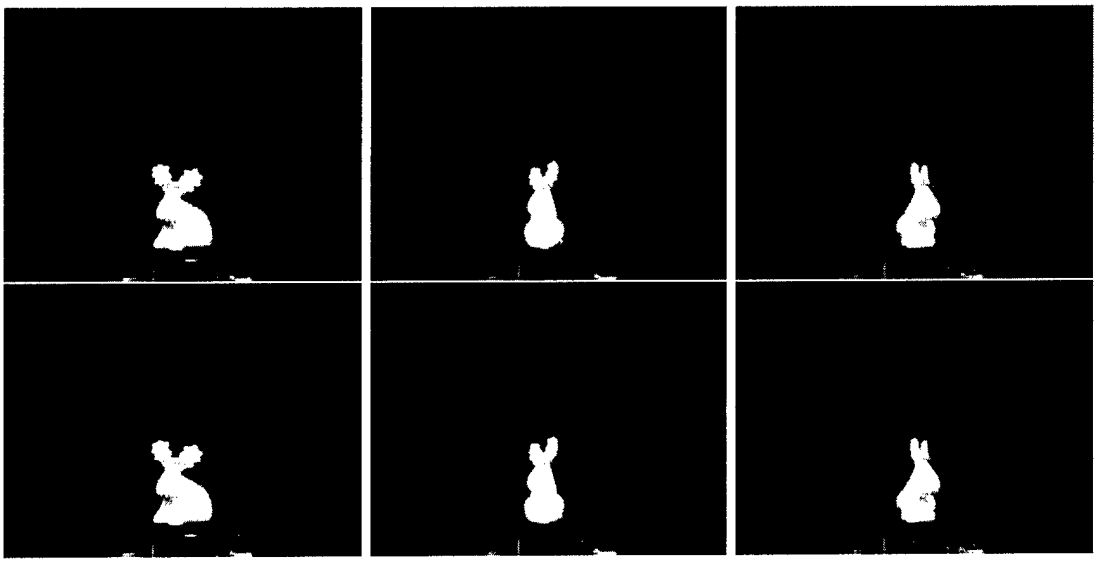


Figure 3.3: Background segmentation results using thresholding of a few reindeer piggy bank images.

voxel on the same plane is distributed among all the available processors in our parallel implementation of space carving. We accomplish this by using Pragma compiler directives for parallel code execution in C.

3.5 Experimental Results

To evaluate the performance of our 3D reconstruction system, we tested the program on several objects. For all the different experiments, we used grayscale images. We also used a 20% standard deviation threshold of the grayscale values to determine whether or not the voxels should be declared photo-inconsistent and consequently carved. This relatively high threshold was chosen in order to compensate for illumination effects and errors in calibration. In all experiments, the program was run on the Onyx R10000 supercomputer using 20 processors.

In the first experiment, we captured 36 images of the toy Barney at 10-degree angle increments in order to generate a complete view of the object. The initial volume was discretized into 70x70x70 voxels for a total of 343,000 voxels. The object is reconstructed after five passes of the space carving algorithm, and a total time of 1 minute and 5.98 seconds. The final volume contained 4,228 voxels. Figure 3.4 shows a few input images and the reconstructed result.

As we can see, the shape of Barney was reconstructed quite accurately, although some fine details of the toy's texture were lost. This can be explained by the fact that the initial volume was not discretized at a high enough resolution.

Discretizing the initial scene volume into a larger number of voxels, which would result in a decrease of the size of the voxels, should allow us to capture finer details. To prove this, we ran the program with the scene volume initialized at 150x150x150 voxels (3,375,000 voxels),

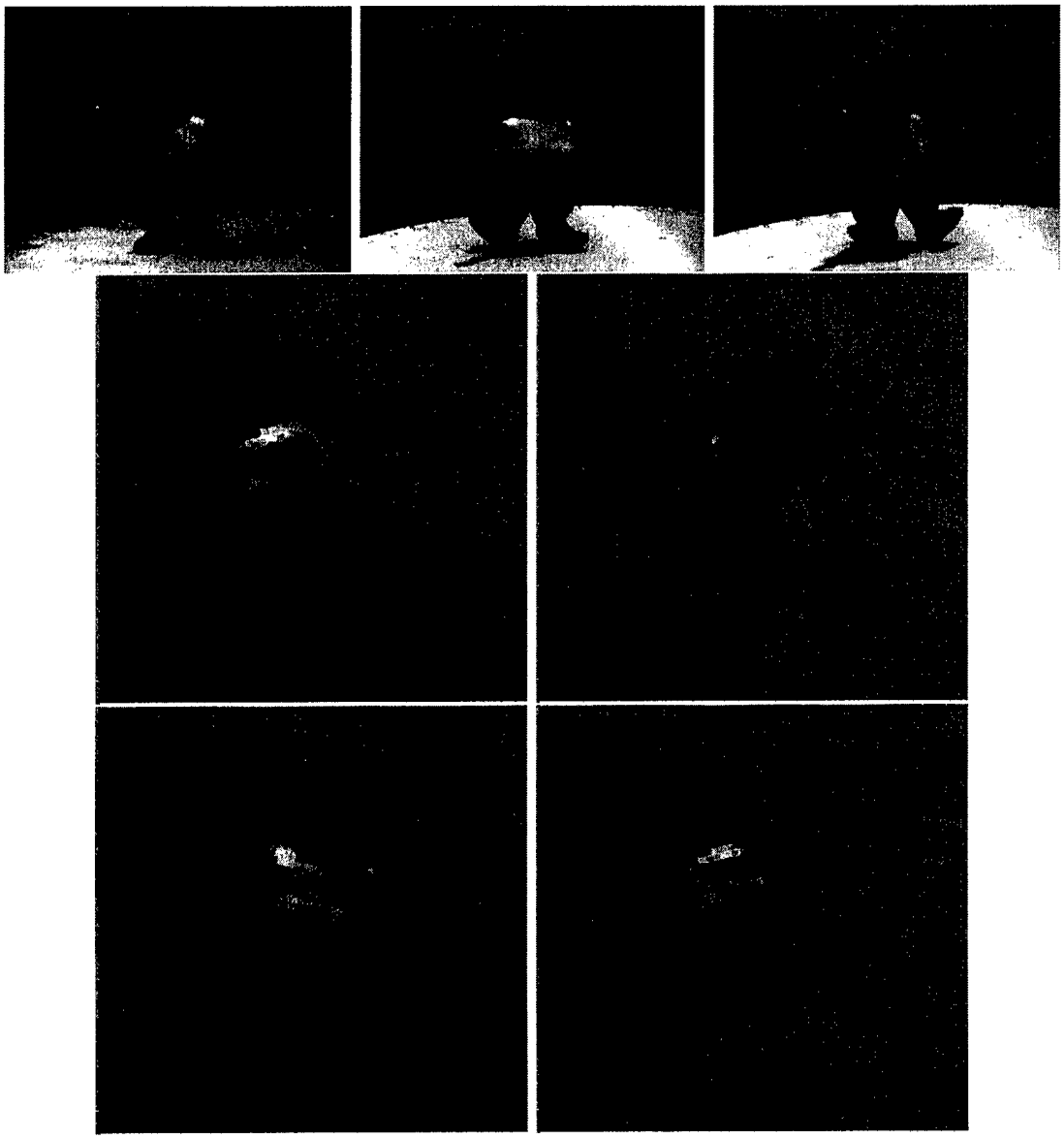


Figure 3.4: A few examples of the images captured for the Barney toy reconstruction. Reconstructed model with a volume space initialized at 70×70×70 is also shown.

and then at 270x270x270 voxels (19,683,000 voxels). Table 3.1 shows us the execution time and the final number of voxels at the different discretization levels.

As we can see from Figure ?? and Figure ??, more details are captured in our 3D reconstructed model as the volume resolution increases. However, there is a price to pay. Finer reconstruction comes at the expenses of processing time and memory.

Next, we attempted to judge the improvement in reducing the execution time gained by parallelizing space carving. All the reconstuctions mentioned above were conducted in a parallel mode using 20 processors. Using the thirty-six images of the toy Barney that

Table 3.1: Reconstruction statistics for the Barney toy with volume space initialized at 70×70×70, 150×150×150 and 270×270×270 voxels.

Number of input images	**Volume Space**	**Initial number of voxels**	**Final number of voxels**	**Processing time**
36	70x70x70	343,000	4,228	1mn 5.98secs
36	150x150x150	3,375,000	21,054	17mns 20.23secs
36	270x270x270	19,683,000	68,008	2hrs 59mns 47.02secs

we captured in the previous experiment, we performed the reconstruction using a single processor (serial mode) with volume space initialized at 100x100x100. We also performed the same reconstruction using 5, 10 and 15 processors in order to analyze the program execution speed gained by using more processors. The results are tabulated in Table 3.2.

Table 3.2: *Space carving program execution times using various numbers of processors.*

Number of Processors	**Number of Images**	**Initial number of voxels**	**Processing time**
1	36	1,000,000	37mns 18.38secs
5	36	1,000,000	8mns 22.51secs
10	36	1,000,000	4mns 33.08secs
15	36	1,000,000	3mns 12.16secs
20	36	1,000,000	2mns 43.28secs

As we can see from these results, there is a significant advantage to our parallel implementation of space carving. Running the program serially (on a single processor) takes a considerable amount of time. Our parallel version of space carving allows a significant reduction in program execution time by distributing the workload to n processors. By using a larger number of processors in the reconstruction, we can considerably scale down the program execution time.

In all experiments, calibrated images were successfully acquired through the setup. From the images, the space carving program succeeded in reconstructing the shape of the objects regardless of the positions of the camera. The 3D models were generated in reasonable amount of times on the Onyx R10000 supercomputer. The program could be easily scaled up to run on more processors to further speed up its execution time to the capabilities of a more powerful computing machine. In order to provide the user with a visual measure of the quality of the reconstructions, all the reconstructed models were displayed on the Imersa Desk, a stereo visualization screen at the computer vision and Image Processing (CVIP) laboratory.

Chapter 4

The CardEye: A Trinocular Active Vision system

Active vision employs controlled changes in the acquisition process to obtain better information for the solution of computational vision problems. Early work on active vision can be attributed to Tenenbaum, who introduced the term "image inadequacy" to describe the limitations of the static imaging framework for recovery problems [42]. Examples of active vision can be found in the work of Bajcsy, Ballard, Aloimonos, and Abbott and Ahuja [1, 6].

The active vision area has been enriched by the design and construction of a variety of active vision platforms and the use of different active vision techniques to improve the machine perception. For references and descriptions see [8, 45]. Noteworthy are Yorick (University of Oxford) [37], Rochester head (University of Rochester) [7], FOVEA (University of Texas) [22], PennEyes (University of Pennsylvania) [28], BiSight (HelpMate Robotics Inc.) [47] and INRIA head (INRIA, France) [46]. Many of these are one-of-a-kind prototypes, using only two cameras to mimic some of the components and the functionality of the human vision system.

The CardEye system is an attempt to mimic the functionality of the human vision system without being restricted to its components. Thus, CardEye utilizes more sensors than human beings, which improves the resultant performance. Specifically, the system has three cameras to improve the recovery process, and the system uses an active lighting device to assist in surface reconstruction process. Moreover, the system employs active vision techniques to improve the machine perception. The system has the basic mechanical properties of active vision platforms - pan, tilt, roll, focus, zoom, aperture, vergence and baseline. The flexibility of the system and the availability of different sensors will assist in solving many problems in active vision research. In this paper, we describe the architecture of the system and our ongoing research on its functionality. More details can be found in [20].

4.1 The Mechanical Design

The aim of the CardEye project is to build an active vision system using a trinocular head that can possess basic mechanical properties such as pan, tilt, roll, focus, zoom, aperture, vergence and baseline. Building a trinocular system with these properties adds more complexity

and redundancy to the system. To eliminate the redundancy, we assigned the mechanical properties to the system as a whole and not to each camera. As a consequence, the three cameras are coupled together to perform the same motion, to fixate to a point, or to change the baseline while a robotic arm, on which the camera assembly is mounted, provides the flexibility to pan, tilt, or roll. Of course, active lenses have the zoom and focus properties. The active lighting device consists of a laser generator, different diffractor filters mounted on a rotating drum, and a mounting mechanism that enables the device to change its orientation around the system vertical axis and to switch the filter. With the help of this device, structured-light techniques for surface reconstruction can be easily utilized in our system.

A simulated design, shown in Fig. 4.1, has been used to test the coordination between the system parts, to justify the available degrees of freedom and to justify the solution of the system kinematics with different target locations. As shown in Fig. 4.1, the system consists of a trinocular head holding three cameras at equal distance from each other and an active lighting device mounted at the center of the trinocular head. The cameras can translate along their mounts to change the baseline distance. At the same time, the cameras can rotate towards each other to fixate to a point in space. This is known as the vergence property. The trinocular head is connected to a robotic arm with at least three segments and four joints - the base, elbow, shoulder and wrist. The base joint provides the pan property. The shoulder and elbow joints provide the tilt property. The wrist joint provides the roll property.

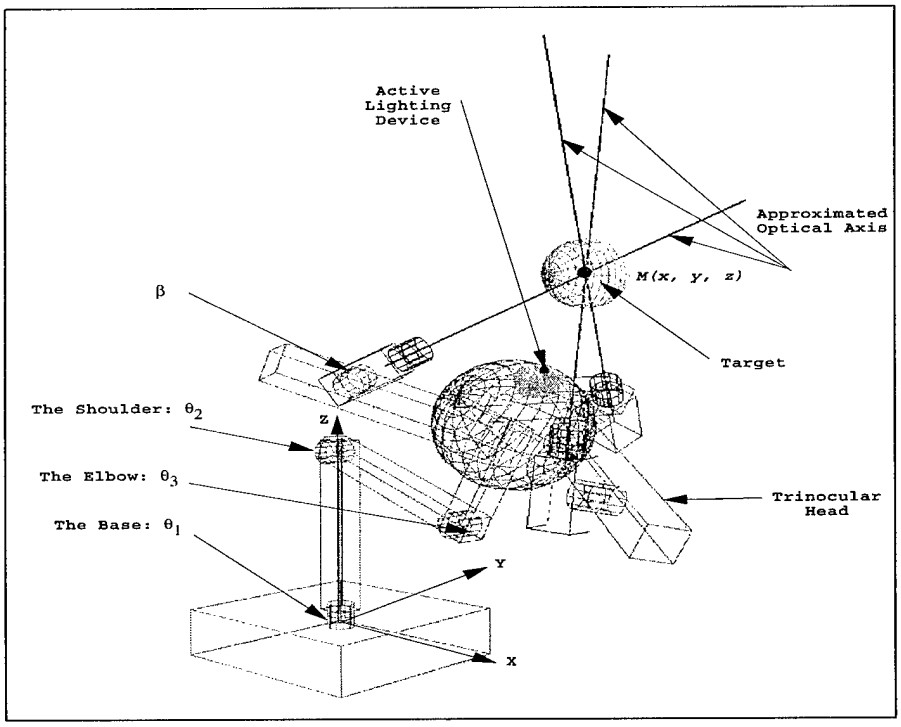


Figure 4.1: CardEye simulated design (A trinocular head attached to three-segment robotic arm.)

In contrast to some vision heads which are based on the isosceles right-angle model (e.g., [31]) to simplify the stereo matching process (but the mechanical difficulties of the alignment of that model are rather difficult to overcome), the CardEye's camera configuration mimics an important property of the human vision system, which is known as cyclopean view. It has been known for some time (Hering 1897, Ibn Al-Haytham around 1000) [21] that under normal viewing conditions, the world appears to us as seen from a virtual eye placed midway between the left and right eye positions. The geometry of this cyclopean eye is depicted in Fig. 4.2. As shown in the binocular system, the cyclopean eye/camera fixates to the same fixation point of the actual cameras. The optical axis of the cyclopean camera is the bisector of the optical axes of the real cameras. By analogy, the trinocular system is constructed as shown in the figure.

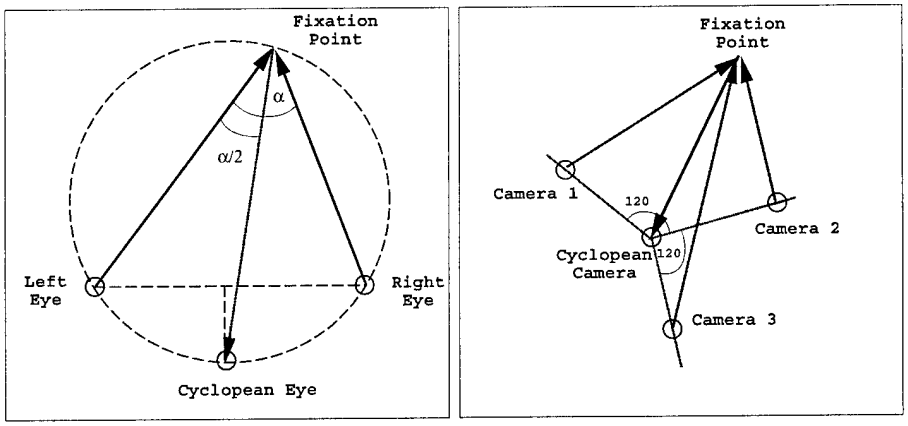


Figure 4.2: The cyclopean view in binocular and trinocular vision systems.

The fixation process in the system is performed in two steps. The first step is to change the robotic arm joints to align the fixation point with the end-effector segment of the arm. The second step is to rotate and translate the cameras to fixate at a specific point. The fixation process is illustrated using a schematic diagram of the system (Fig. 4.3). As shown in the schematic diagram, the system parameters are denoted as follows:

- t corresponds to the distance from a camera to the center of the head,
- β corresponds to the vergence angle,
- θ_1 corresponds to the pan angle,
- θ_2 and θ_3 correspond to the tilt angle,
- θ_4 corresponds to the roll angle,
- d is the distance along the fixation line between the fixation point and the cameras plane.

The complete solution of the system kinematics is described next.

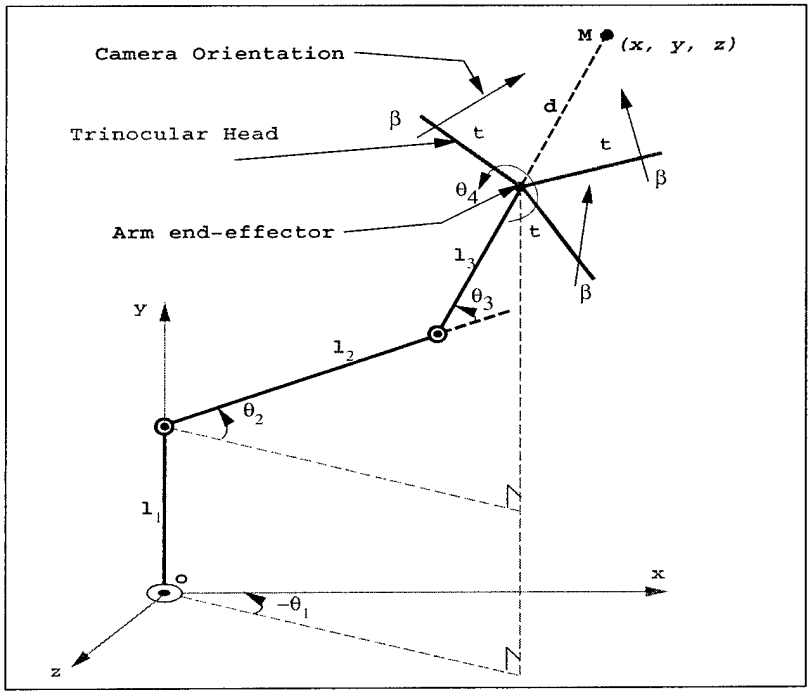


Figure 4.3: CardEye schematic diagram (The trinocular head attached to the end-effector of the 3-segment robotic arm. **M** is a target fixation point.)

4.1.1 The System Kinematics

The CardEye system consists of two parts, each is considered a manipulator by itself. The first manipulator is a three-link robotic arm that provides three degrees of freedom (pan, tilt and roll) similar to the human neck. The other manipulator is the trinocular head that provides the vergence property similar to the human eyes. It provides a variable baseline as well. We shall discuss the kinematics of each section separately.

4.1.1.1 The Robotic Arm Kinematics

The problem of inverse kinematics is posed as follows: given the position and the orientation of the end-effector of the manipulator, calculate all possible sets of joint angles which could be used to attain this given position and orientation. The solution to the inverse kinematics problem can be approached either numerically [32] or analytically [34]. The analytical approaches [34] exploit the specific geometry of the manipulator and determine a closed-form expression of the solution. Therefore, we use the analytical method to solve the kinematics problem of the robotic arm.

As shown in Fig. 4.4, let oa, ab and bM be the first, second and third arm segments, respectively and θ_1, θ_2 and θ_3 the base, shoulder and elbow angles, respectively. Let m be the projection of $M = (x,\ y,\ z)$ in the $x - z$ plane, hence the triangle oem is a right-angle

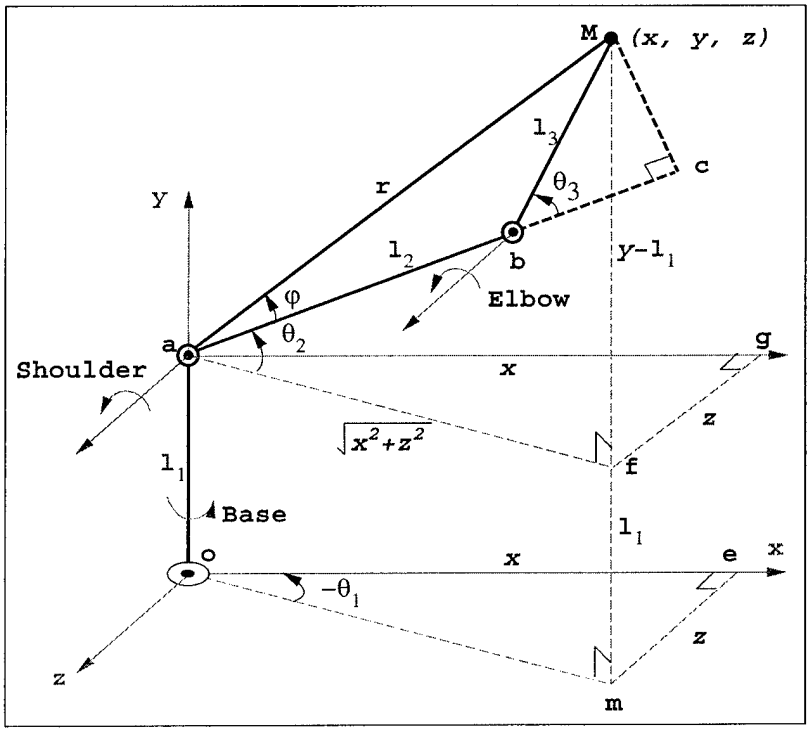


Figure 4.4: Inverse kinematics for three-link arm.

triangle and thus the angle θ_1 can be expressed as:

$$\theta_1 = -\tan^{-1}(z/x) \tag{4.1}$$

Before computing the shoulder and elbow angles, we test the existence of solutions. From the triangle abM we found that there are three cases. A unique solution will be extracted if $l_2 + l_3 = r$. There is no solution for the cases in which $l_2 + l_3 > r$. There are two solutions if $l_2^2 + l_3^2 \leq r^2$. The one shown in Fig. 4.4 is the "elbow down" solution. Another solution may be determined for the "elbow up" configuration where both links are above the vector $\overrightarrow{aM}$. r can be computed using the triangle agf and afM. From the right angle triangle agf we get

$$|\overrightarrow{af}| = \sqrt{x^2 + z^2} \tag{4.2}$$

From the right angle triangle afM we get,

$$r^2 = x^2 + (y - l_1)^2 + z^2 \tag{4.3}$$

Assuming that a solution exists then r can be obtained from the right angle triangle acM as follows

$$r^2 = (l_2 + l_3 cos(\theta_3))^2 + (l_3 sin(\theta_3))^2 \tag{4.4}$$

We could solve Eq. 4.4 and 4.3 for θ_3 using the inverse cosine function. However, it is better

to use the inverse tangent for numerical accuracy. Therefore, we proceed by computing

$$\begin{aligned} \cos\ (\theta_3) &= \frac{r^2 - l_2^2 - l_3^2}{2l_2l_3} = C \\ \sin\ (\theta_3) &= \pm\sqrt{1 - \cos^2(\theta_3)} = \pm\sqrt{1 - C^2} = D \\ \theta_3 &= \tan^{-1}(D/C) \end{aligned} \tag{4.5}$$

To determine θ_2, we define the auxiliary angle ϕ in the figure. By inspection of the right angle triangles acM and afM, we obtain

$$\theta_2 = \tan^{-1}(\frac{y - l_1}{\sqrt{x^2 + z^2}}) - \tan^{-1}(\frac{l_3 \sin(\theta_3)}{l_2 + l_3 \cos(\theta_3)}). \tag{4.6}$$

4.1.1.2 The Trinocular Head Kinematics

The three cameras of the trinocular head have the capability to converge towards a point along a line perpendicular to the plane that passes through the midpoint of the trinocular head. This line is known as the fixation line. The fixation line in the system represents the trajectory of the fixation points of the cameras. Fig. 4.5 shows a schematic diagram of the trinocular head. The default orientation of the three cameras is parallel to the fixation line. The three cameras converge to the fixation line by angle of rotation $\beta = \tan^{-1}(t/d)$.

Given a target object (accordingly a fixation point), the optimal value of t is determined

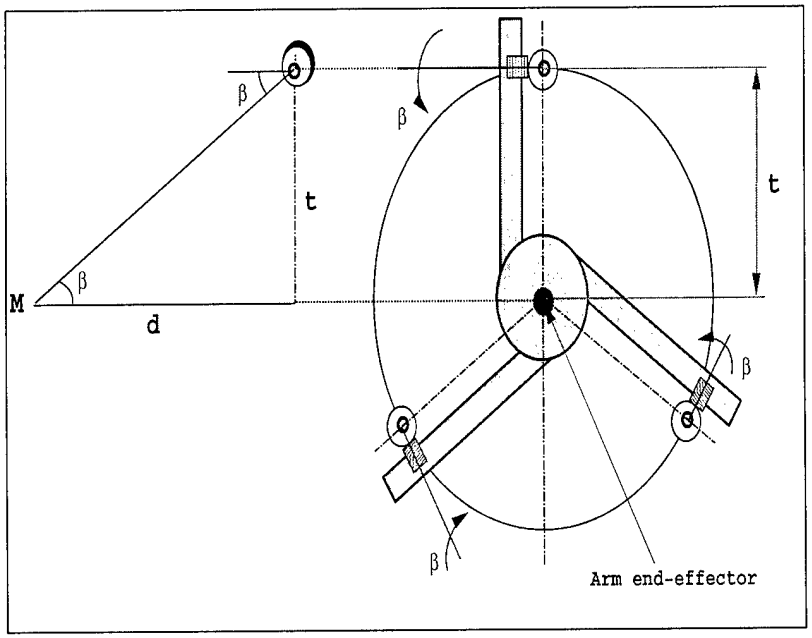


Figure 4.5: The trinocular head kinematics.

by the sensor planning module (described later) to fulfill some system requirements. Once t is known, the angle of rotation for the three cameras, β, can be computed.

4.1.2 System Integration

In the current phase of the system, the head has been fully implemented, while the robotic arm is left for the next phase. Fig. 4.6 shows the CardEye system and the cabinet that encompasses all the digital circuitry interfacing the system to a network of high-end workstations, PCs, an Onyx R10000 supercomputer and an ImmersaDesk visualization screen.

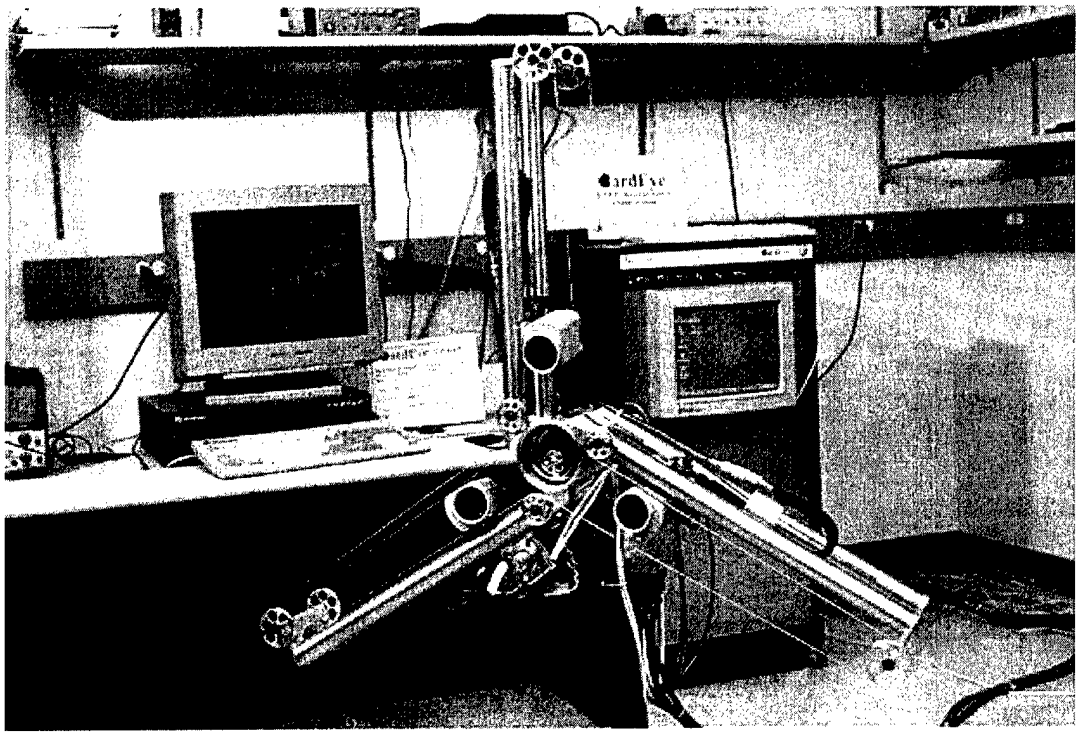


Figure 4.6: A picture of the CardEye system and the control circuitry cabinet.

4.2 The System Functionality

An overview of CardEye functionalities is given in Fig. 4.7. The figure shows four main modules: the sensor planning, camera calibration, surface reconstruction, and decision making. The fourth function, decision making, uses the output of the reconstruction module to specify the next fixation point of the system. The application for which CardEye will be used is the basis for the decision. For each application, we define a mode of operation such as 3D model building, object tracking, object recognition and navigation. In this work, we briefly describe the functionality of the system for 3D object reconstruction. To achieve this task, several features of the system are novel:

- a sensor planning module [24] that solves for system parameters that maximize the effectiveness of the reconstruction process,

- a camera calibration technique [2] that can capture the variation of the camera model parameters as continuous functions of lens settings (zoom, focus and aperture),
- a multi-stage reconstruction approach [19, 18] that combines structured-light, edge-based stereo and area-based stereo reconstruction techniques.

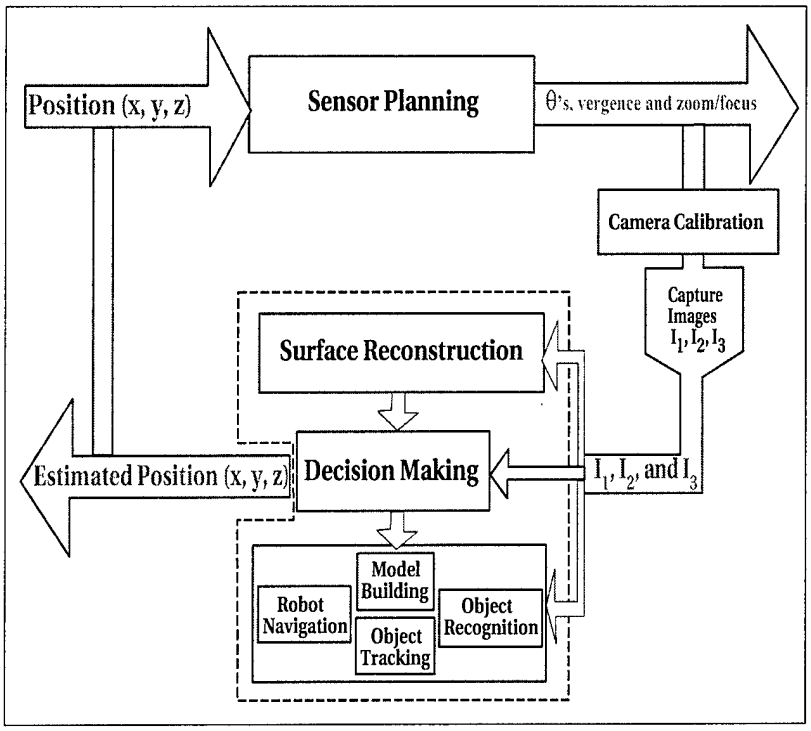


Figure 4.7: The CardEye functionality.

4.2.1 Sensor Planning

The sensor planning process [24] employs the fixation point, generated by the decision making process, to generate the sensor parameters (pan, tilt, roll, vergence, baseline, zoom and focus) satisfying different vision constraints (field of view, focus, disparity and overlap).

The goal for the sensor planning component is to maximize the effectiveness of the 3D reconstruction algorithm from one frame. For effective reconstruction, the frames must display adequate depth information and have a fairly large overlap area. The two goals are in conflict. For more overlapped area, the converged cameras are to move closer, while better disparity content requires the cameras to move away from each other. Consequently, the algorithm solves for the translation between the cameras, t, that satisfies both goals to a certain extent. Moreover, the sensor planning algorithm makes sure that the target object is within the field of view of the head and in focus. As a result, the vergence angle, cameras' zoom and focus settings are determined. The interested reader is referred to [24],[25] for further details.

4.2.2 Zoom-lens Camera Calibration

The CardEye uses three zoom-lens cameras, which need to be calibrated to know the camera model parameters at any lens setting (zoom and focus) as determined by the sensor planning module. Camera systems with automated zoom-lenses are inherently more useful than those with fixed-parameter (*passive*) lenses due to their flexibility and controllability. In such cameras, the image-formation process varies with the lens optical settings, thus many of the camera model parameters are non-linear functions of the lens settings. The calibration problem of these cameras relies on formulating functions that describe the relationships between the camera model parameters and the lens settings. As opposed to passive cameras, this raises several challenges [49]. To solve this difficult task, we developed a neural framework [2] based on our novel neurocalibration approach [3], which cast the classical geometric (passive) camera calibration problem into a learning problem of a multi-layered feedforward neural network (MLFN). This framework consists of a number of MLFNs learning concurrently, independently and cooperatively, to capture the variations of model parameters across lens settings.

This framework offers a number of advantages over other techniques (e.g., [49],[38]): it can capture complex variations in the camera model parameters, both intrinsic and extrinsic (as opposed to polynomials in [49]); it can consider any number/combination of lens control parameters, e.g., zoom, focus and/or aperture; all of the model parameters are fitted to the calibration data in a global optimization stage at the same time while minimizing the calibration error.

4.2.3 Surface Reconstruction

Due to the different characteristics of object surfaces in the environment, a single recovery technique does not work well in all situations, and thus we employ multiple recovery techniques to reconstruct a 3D map for the same scene [19], that was described in Chapter 2. This technique integrates edge-based stereo and area-based stereo to combine the accuracy of the former and the richness of the latter and employs structured light to reconstruct featureless and smooth objects that cannot be properly handled with edge- or area-based stereo. The integration is performed by reconstructing the actual and induced edges in the scene using the geometrical constraints of trinocular vision [18], followed by applying a correlation-based technique to fill the gaps between the reconstructed features. Fig4.8 shows the reconstruction results of some objects. Simple objects (e.g., boxes) are, also, used to evaluate the reconstruction process and the *rms* error between the reconstructed object dimensions and the ground truth values is within 5 millimeters. More details about the approach and performance analysis of the system can be found in [20].

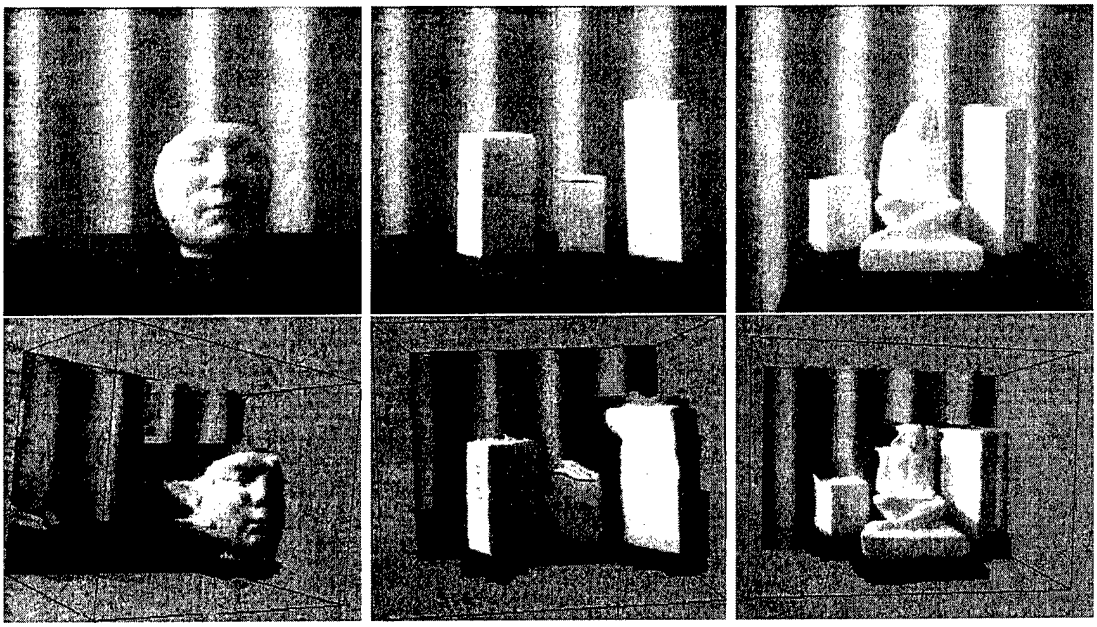


Figure 4.8: Reconstruction results from the CardEye: original images are shown in the first row and the cyclopean view of the reconstructions after adding texture are shown in the second row.

Bibliography

[1] A. L. Abbott and N. Ahuja. Active surface reconstruction by integrating focus, vergence stereo and camera calibration. *ICCV*, 1990.

[2] M. Ahmed and A. Farag. A neural optimization framework for zoom-lens camera calibration. *CVPR*, 2000.

[3] M. Ahmed, E. Hemayed, and A. Farag. Neurocalibration: a neural network that can tell camera calibration parameters. *ICCV*, 1999.

[4] N. Ayache and F. Lustman. Trinocular stereo vision for robotics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(1):73–85, Jan. 1991.

[5] N. J. Ayache and O. D. Faugeras. Building, registrating, and fusing noisy visual maps. *International Journal of Robotics Research*, 7(6):45–65, Dec. 1988.

[6] R. Bajcsy. Active perception vs. passive perception. *Proceedings: Workshop on Computer Vision*, 1985.

[7] C. M. Brown. The rochester robot. 1988.

[8] H. I. Christensen and J. L. Crowley. *Active Vision Systems.* Kluwer Academic Publishers, Amsterdam, 1996.

[9] W. Culbertson, T. Malzbender, and G. Slabaugh. Generalized voxel coloring. *Vision Algorithms 99 Workshop*, pages 67–74, Sept. 1999.

[10] F. W. DePiero and M. M. Trivedi. 3-d computer vision using structured light: Design, calibration and implementation issues. *Advances in Computers*, 43, 1996.

[11] O. Faugeras et. al. *Real time correlation-based stereo: algorithm, implementations and applications.* Technical Report, RR-2013, 1993.

[12] O. Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint.* the MIT Press, 1993.

[13] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. *Proceedings of Fifth European Conference On Computer Vision*, June, 1988.

[14] P. Fua and Y.G. Leclerc. Object-centered surface reconstruction: combining multi-image stereo and shading. *Int. Journal of Computer Vision*, 16(1):35–56, 1995.

[15] P. Fua and Y.G. Leclerc. From multiple stereo views to multiple 3d surfaces. *Int. Journal of Computer Vision*, 24(1):19–35, 1997.

[16] W. E. L. Grimson. A computer implementation of a theory of human stereo vision. *Philos. Trans. R. Soc. London, Ser. B*, 292:217–253, 1981.

[17] E. Hemayed, Moumen Ahmed, and A. Farag. Cardeye: A 3d trinocular active vision system. *Proc. 3rd IEEE Conference on Intelligent Transportation Systems (ITSC'2000)*, pages 398–403, Oct. 2000.

[18] E. Hemayed and A. Farag. A geometrical-based trinocular vision system for edges reconstruction. *ICIP*, 1998.

[19] E. Hemayed and A. Farag. Integrating edge-based stereo and structured light for robust surface reconstruction. *Int. Conf. Intelligent Vehicles*, 1998.

[20] E. E. Hemayed. *A 3D Trinocular Active Vision System for Surface Reconstruction.* Ph.D. Thesis, CVIP Lab, University of Louisville, August 1999.

[21] I. P. Howard and B. J. Rogers. *Binocular Vision and Stereopsis.* Oxford University Press, 1995.

[22] W. N. Klarquist and A. C. Bovik. Fovea: A foveated vergent active stereo vision system for dynamic three-dimensional scene recovery. *IEEE Trans. Robotics and Automation*, 14(5), October 1998.

[23] K. Kutulakos and S. Seitz. A theory of shape by space carving. *Proc. of IEEE Int. Conf. on Computer Vision*, pages 307–314, Sept. 1999.

[24] P. Lehel, E. E. Hemayed, and A. A. Farag. Sensor planning for a trinocular active vision system. *CVPR*, 1999.

[25] P. Lehel, E. E. Hemayed, and A. A. Farag. Sensor planning for a trinocular active vision system. *submitted to PAMI*, 1999.

[26] S.D. Ma, S.H. Si, and Z.Y. Chen. Quadric curve based stereo. *Proceedings. 11th IAPR International Conference on Pattern Recognition*, 1:xxv+795, Sept. 1992.

[27] Hans-Gerd Maas. Robust automatic surface reconstruction with structured light. *International Archives of Photogrammetry and Remote Sensing*, XXIX (B5):709–713, 1992.

[28] B. C. Madden and U. C. Seelen. *PennEyes: A binocular active vision system.* Technical Report, University Of Pennsylvania, 1995.

[29] P. J. Narayanan, P. Rander, and T. Kanade. Constructive virtual worlds using dense stereo. *Proc. of IEEE Int. Conf. on Computer Vision (CVPR)*, pages 3–10, Jan. 1998.

[30] Robert L. Ogniewicz. *Automatic Medial Axis Pruning.* Technical Report no. 95-4, Harvard Robotics Lab., 1995.

[31] Y. Ohta, M. Watanabe, and K. Ikeda. Improving depth map by right-hand trinocular stereo. *ICPR*, pages 519–521, 1986.

[32] R. P. Paul. *Robot Manipulators: Mathematics, Programming and Control.* Cambridge, MA: MIT Press, 1981.

[33] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *Proceedings of IEEE International Conference On Computer Vision*, pages 492–499, Jan. 1998.

[34] R. J. Schilling. *Fundamentals of Robotics: analysis and control.* Prentice Hall, New Jersey, 1990.

[35] W.B. Seales and O.D. Faugeras. Building three-dimensional object models from image sequences. *Computer Vision and Image Understanding*, 61(3):308–324, May 1995.

[36] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1067–1073, 1997.

[37] P. M. Sharkey, D. W. Murray, S. Vandevelde, I. D. Reid, and P. F. McLauchlan. A modular head/eye platform for real-time reactive vision. *Mechatronics Journal*, 3(4):517–535, 1993.

[38] S. W. Shih, Y.P. Hung, and W. S. Lin. *Calibration of an active binocular head*, volume 28(4). IEEE Trans. Man, Sys. and Cybernetics, July 1998.

[39] B.J. Super and W.N. Klarquist. Patch-based stereo in a general binocular viewing geometry. *PAMI*, 19(3), March 1997.

[40] M.J. Swain and M. Stricker. *Promising directions in active vision.* Technical Report CS 91-27, University of Chicago, 1991.

[41] J. P. Tarel and J. M. Vezien. A generic approach for planar patches stereo reconstruction. *Proc. Scandinavian Conference on Image Analysis (SCIA)*, 1995.

[42] J. Tenenbaum. *Accommodation in Computer Vision.* Ph.D. Dissertation, Stanford, 1970.

[43] R. Vaillant and O. D. Faugeras. Using extremal boundaries for 3d object modeling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):157–173, Feb. 1992.

[44] R. J. Valkenburg and A. M. McIvor. *Accurate 3-D measurement using a structured light system.* Technical Report 576, Industrial Research Limited, June 1996.

[45] T. Vieville. *A few steps towards 3D active vision.* Springer-Verlag, Information Sciences Series, 1997.

[46] T. Vieville, E. Clergue, R. Enciso, and H. Mathieu. Experimenting with 3-d vision on a robotic head. *Robotics and Autonomous Systems*, 14(1), Jan/Feb 1995.

[47] C. Weiman and M. Vincze. A generic motion platform for active vision. *SPIE Conf 2904*, Nov 1996.

[48] W. Wen and B. Yuan. Stereo correspondence for planar curves based on their invariant. *Proc. Europe - China Workshop*, 403:230–237, April 1995.

[49] R. G. Wilson. *Modeling and calibration of automated zoom lenses.* PhD dissertation, CMU, 1994.

[50] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition.* Kluwer Academic Publishers, 1996.

[51] X. Yingen, D. Wang, and Z. Guangzhao. Integrated method of stereo matching for computer vision. *Proc. SPIE - Conf 2847*, pages 665–676, Aug. 1996.